AMERICAN NATIONAL STANDARD

ANSI/IEEE Std 802.2-19
ISO/DIS 8802/2

LOCAL AREA NETWORKS

802.2

ANSI/IEEE Standard
Draft International Standard

Logical Link Control

Published by
The Institute of Electrical and
Electronics Engineers, Inc

Distributed in cooperation with
Wiley-Interscience, a division
of John Wiley & Sons, Inc

An American National Standard

IEEE Standards for Local Area Networks:

Logical Link Control

Published by
The Institute of Electrical and Electronics Engineers, Inc

Distributed in cooperation with
Wiley-Interscience, a division of John Wiley & Sons, Inc

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least once every five years for revision or reaffirmation. When a document is more than five years old, and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
345 East 47th Street
New York, NY 10017
USA

ANSI/IEEE Std 802.2-1985
ISO Draft
International
Standard 8802/2

An American National Standard

IEEE Standards for Local Area Networks:

Logical Link Control

Sponsor

Technical Committee Computer Communications of the IEEE Computer Society

Approved July 16, 1984

IEEE Standards Board

Approved December 28, 1984

American National Standards Institute

The portions of this standard covering type 1 class 1 service have been adopted for U.S. federal government use.

Details concerning its use within the federal government are contained in FIPS PUB 107, Local Area Networks: Base Band Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Specifications and Link Layer Protocol. For a complete list of the publications available in the FEDERAL INFORMATION PROCESSING STANDARDS Series, write to the Standards Processing Coordinator, Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899, U.S.A.

Second Printing (8½″ × 11″) for NTIS

ISBN 0-471-82748-7

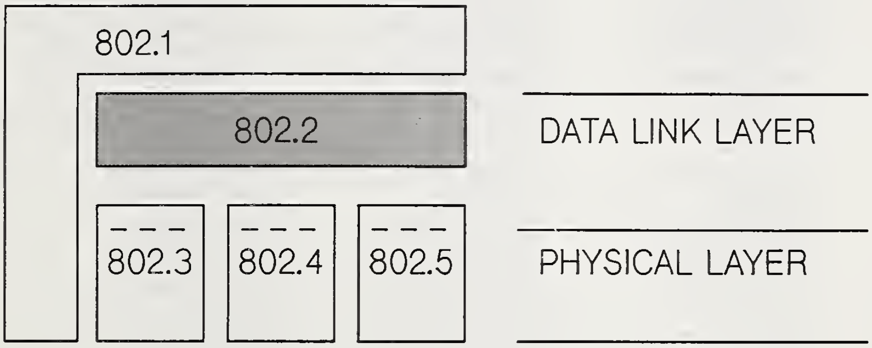
Library of Congress Catalog Number 84-43095

December 17, 1984 *SH09712*

Foreword

(This Foreword is not a part of ANSI/IEEE Std 802.2-1985, IEEE Standard Logical Link Control.)

This standard is part of a family of standards for Local Area Networks (LANs). The relationship between this standard and other members of the family is shown below. (The numbers in the figure refer to IEEE Standard numbers.)



This family of standards deals with the physical and data link layers as defined by the ISO Open System Interconnection Reference Model. The access standards define three types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. The standards defining these technologies are

(1) ANSI/IEEE Std 802.3-1985 (ISO DIS 8802/3), a bus utilizing CSMA/CD as the access method

(2) ANSI/IEEE Std 802.4-1985 (ISO DIS 8802/4), a bus utilizing token passing as the access method

(3) ANSI/IEEE Std 802.5-1985,[1] a ring utilizing token passing as the access method.

Other access methods (for example, metropolitan area networks) are under investigation.

ANSI/IEEE Std 802.2 (ISO DIS 8802/2), IEEE Standard Logical Link Control protocol, is used in conjunction with the medium access standards.

A companion document, IEEE 802.1[2] describes the relationship among these standards and their relationship to the ISO Open System Interconnection Reference Model in more detail. This companion document will contain internetworking and network management issues.

The reader of this standard is urged to become familiar with the complete family of standards.

---

[1] Publication anticipated for early 1985.

[2] In preparation.

At the time of approval of this standard the following members were participants of IEEE Project 802 Working Group.

Om Agrawal
Phil Arneth
Jeff Bobzin
Mark Bauer
Le Biu
Clyde Boenke
Bob Bowen
Bob Bridge
Chuck Brill
Wayne Brodd
Werner Bux
Jim Campbell
Tony Capel
Dave Carlson
Ron Cates
Rao Cherukuri
Po Chen
Jade Chien
Mike Clader
Jerry Clancy
Rich Collins
Steve Cooper
Bob Crowder
Kirit Davé
John Davidson
Em Delahostria
Jan Dolphin
Bob Donnan
Bob Douglas
Bill Durrenberger
Rich Fabbri
Eldon Feist
Jim Field
Larry Foltzer
Ron Floyd
Darrell Furlong
Mel Gable
Mike Garvey
Bud Glick
Arie Goldberg
Pat Gonia
Gordon Griffiths
Bob Grow
Maris Graube
Ed Harada
Lo Hsieh
Karen Hsing
Kevin Hughes
Marco Hurtado
Bob Husak
Dittmar Janetzky
Ross Jaibaji
George Jelatis
Gabor Kardos
Peggy Karp
Kristin Kocan
Zak Kong
Sy Korowitz
George Koshy
Don Kotas
Tony Kozlik
Mike Kryskow
Dave Laffitte
Terry Lawell
Ron Leuchs
Peter Lin
Jim Lindgren
Laurie Lindsey
Bill Livingston
Then Tang Liu
Don C. Loughry
Don J. Loughry
Bruce Loyer
Jerry Lurtz
Bill Miller
Ken Miller
Lou Mitta
Bob Moles
Jim Mollenauer
Ware Myers
Gene Nines
Bill Northup
Brian O'Neil
Kul Padda
Mahendra Patel
Tom Phinney
Juan Pimentel
Lavern Pope
Dave Potter
Dennis Quy
John Rance
Dan Ratner
Richard Read
Ted Rebenko
John Ricketson
Edouard Rocher
Rob Rosenthal
Chip Schnarel
Walter Schreuer
Gerard Segarra
Dennis Sosnoski
Robert C. Smith
Mark Stahlman
Steve Stearns
Garry Stephens
Mark Steiglitz
Kathleen Sturgis
Bob Stover
Bart Stuck
Dave Sweeton
Dan Sze
Vic Tarassov
Angus Telfer
Dave Thompson
Fouad Tobagi
Jean-Marie Tourret
Bo Viklund
Bruce Watson
Don Weir
Dan Wendling
Walter Wheeler
Hugh White
Steve Whiteside
Earl Whitaker
Ping Wu
Esin Ulug
Hiroshi Yoshida
Hank Zannini

Contents

SECTION PAGE

FIGURES

TABLES

An American National Standard

IEEE Standards for Local Area Networks:

Logical Link Control

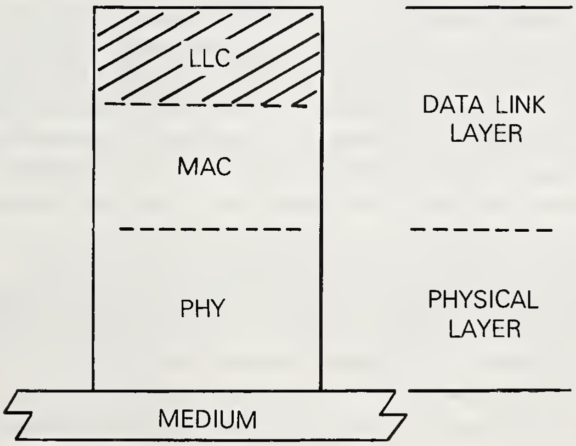
1. Introduction

1.1 Scope and Purpose. This standard is one of a set of standards produced to facilitate the interconnection of computers and terminals on a local area network (LAN). It is related to the other standards by the Reference Model for Open Systems Interconnection.

NOTE: The exact relationship of the layers described in this standard to the layers defined by the OSI Reference Model is under study.

This standard describes the functions, and features and protocol of the Logical Link Control (LLC) Sublayer in the IEEE/Std 802 Local Area Network Protocol. The LLC sublayer constitutes the top sublayer in the Data Link Layer (see Fig 1-1) and is common to the various medium access methods that are defined and

Fig 1-1
Relationship to LAN Reference Model



supported by the IEEE/Std 802 activity. Separate standards describe each medium access method individually and indicate the additional features and functions that are provided by the Medium Access Control (MAC) Sublayer in each case to complete the functionality of the Data Link Layer as defined in the 802 architectural reference model.

This standard describes the LLC Sublayer Interface Service Specifications to the Network Layer (Layer 3), to the MAC Sublayer, and to the LLC Sublayer Management function. The interface service specification to the Network Layer provides a description of the various services that the Logical Link Control Sublayer, plus underlying layers and sublayers, offer to the Network Layer, as viewed from the Network Layer. The interface service specification to the MAC Sublayer provides a description of the services that the LLC Sublayer requires of the MAC Sublayer. These services are defined so as to be independent of the form of the medium access methodology, and of the nature of the medium itself. The interface service specification to the LLC Sublayer Management function provides a description of the management services that are provided to the LLC Sublayer. All of the above Interface Service Specifications are given in the form of primitives that represent in an abstract way the logical exchange of information and control between the LLC Sublayer and the identified service function (Network Layer, MAC Sublayer or LLC Sublayer Management function). They do not specify or constrain the implementation of entities or interfaces.

This standard provides a description of the peer-to-peer protocol procedures that are defined for the transfer of information and control between any pair of Data Link Layer service access points on a local area network. The LLC Procedures are independent of the type of medium access method used in the particular local area network.

To satisfy a broad range of potential applications, two types of data link control operation are included (see Section 4). The first type of operation (see Section 6) provides a data-link-connectionless service across a data link with minimum protocol complexity. This type of operation may be useful when higher layers provide any essential recovery and sequencing services so that these do not need replicating in the Data Link Layer. In addition, this type of operation may prove useful in applications where it is not essential to guarantee the delivery of every Data Link Layer data unit. This type of service is described in this standard in terms of "logical data links". The second type of operation (see Section 7) provides a data-link-connection-oriented service across a data link comparable to existing data link control procedures provided in national and international standards such as ADCCP (see 1.3 (3)) or HDLC (see 1.3 (1)). This service includes support of sequenced delivery of Data Link Layer data units, and a comprehensive set of Data Link Layer error recovery techniques. This second type of service is described in this standard in terms of "data link connections."

This standard identifies two distinct "classes" of LLC operation. Class I provides data-link-connectionless service only. Class II provides data-link-connection-oriented service plus data-link-connectionless service. Either class of operation may be supported.

The basic protocols described herein are peer protocols for use in multi-station

multi-access environments. Because of the multi-station, multi-access environment, it shall be possible for a station to be involved in a multiplicity of peer protocol data exchanges with a multiplicity of different stations over a multiplicity of different logical data links and/or data link connections that are carried by a single Physical Layer over a single physical medium. Each unique to-from pairing at the Data Link Layer shall define a separate logical data link or data link connection with separate logical parameters and variables. Except where noted, the procedures described in this chapter shall relate to each Data Link Layer logical data link or data link connection separately and independently from any other logical data link or data link connection that might exist at the stations involved.

1.2 Standards Compatibility. The peer protocol procedures defined in Section 5 utilize some of the concepts and principles, as well as commands and responses, of the balanced link control procedures known as Asynchronous Balanced Mode (ABM), as defined in ISO 7809-1984 and ANSI X3.66-1979. (The ABM procedures provided the basis upon which the CCITT Recommendation X.25 Level 2 LAPB procedures were defined.) The frame structure defined for the Data Link Layer procedures as a whole is defined in part in Section 3 of this standard section and in part in those standards that define the various medium access control (MAC) procedures. The combination of a MAC sublayer address and an LLC sublayer address is unique to each Data Link Layer service access point in the local area network.

NOTE: This division of Data Link Layer addressing space into separate MAC and LLC address fields is not presently a part of any present ISO or ANSI Data Link Layer standard.

1.3 References.

(1) ISO

ISO/DRS 4335 Revised, Information processing systems, Data communications, High-level data link control procedures, Consolidation of elements of procedures

ISO 7809-1984, Information processing systems, Data communications, High-level data link control procedures, Consolidation of classes of procedures

ISO 7498-1984, Information processing systems, Open systems interconnection, Basic reference model

(2) CCITT

Recommendation X.25, Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode on public data networks

Draft Recommendation X.200, Reference model on open systems interconnection for CCITT applications

(3) ANSI

X3.66-1979, American National Standard for Advanced Data Communications Control Procedures (ADCCP)

1.4 Acronyms and Definitions

1.4.1 Acronyms

ABM	Asynchronous Balanced Mode
ACK	ACKnowledge
ADCCP	Advanced Data Communication Control Procedures
ADM	Asynchronous Disconnected Mode
ANSI	American National Standards Institute
C	Command
CCITT	International Telegraph and Telephone Consultative Committee
COMVII	COMmission VII
C/R	Command/Response
DA	Destination Address
DCE	Data Circuit-terminating Equipment
DIS	Draft International Standard
DISC	DISConnect
DM	Disconnected Mode
DSAP	Destination Service Access Point
DTE	Data Terminal Equipment
F	Final
FCS	Frame Check Sequence
FRMR	FRaMe Reject
HDLC	High level Data Link Control
I	Information
I	Information transfer format
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
LAN	Local Area Network
LAPB	Link Access Procedure, Balanced
LLC	Logical Link Control
LSAP	Link layer Service Access Point
LSB	Least Significant Bit
LSDU	Link layer Service Data Unit
M	Modifier function bit
MAC	Medium Access Control
N(R)	Receive sequence Number

N(S)	Send sequence Number
OSI	Open Systems Interconnection
P	Poll
PDU	Protocol Data Unit
P/F	Poll/Final
PHY	PHYsical
R	Response
REJ	REJect
RNR	Receive Not Ready
RR	Receive Ready
S	Supervisory format
S	Supervisory function bit
SA	Source Address
SABME	Set Asynchronous Balanced Mode Extended
SAP	Service Access Point
SSAP	Source Service Access Point
TEST	TEST
U	Unnumbered format
UA	Unnumbered Acknowledgment
UI	Unnumbered Information
V(R)	Receive state Variable
V(S)	Send state Variable
XID	eXchange IDentification

1.4.2 Definitions

For the purpose of this standard the following definitions apply:

accept. The condition assumed by a LLC upon accepting a correctly received PDU for processing.

address fields (DSAP and SSAP). The ordered pair of service access point addresses at the beginning of an LLC PDU which identifies the LLC(s) designated to receive the PDU and the LLC sending the PDU. Each address field is one octet in length.

basic status. An LLC's capability to send or receive a PDU containing an information field.

command. In data communications, an instruction represented in the control field of a PDU and transmitted by an LLC. It causes the addressed LLC('s) to execute a specific data link control function.

command PDU. All PDU's transmitted by a LLC in which the C/R bit is equal to "O."

control field (C). The field immediately following the DSAP and SSAP address fields of a PDU. The content of the control field is interpreted by the receiving destination LLC('s) designated by the DSAP address field:

a. As a command, from the source LLC designated by the SSAP address field, instructing the performance of some specific function;

b. As a response, from the source LLC designated by the SSAP address field.

data link. An assembly of two or more terminal installations and the interconnecting communications channel operating according to a particular method that permits information to be exchanged; in this context the term *terminal installation* does not include the data source and the data sink.

data link layer. The conceptual layer of control or processing logic existing in the hierarchical structure of a station that is responsible for maintaining control of the data link. The data link layer functions provide an interface between the station higher layer logic and the data link. These functions include address/control field interpretation, channel access and command PDU/response PDU generation, transmission and interpretation.

exception condition. The condition assumed by an LLC upon receipt of a command PDU which it cannot execute due to either a transmission error or an internal processing malfunction.

global (broadcast) DSAP address. The predefined LLC DSAP address (all ones) used as a broadcast (all parties) address. It can never be the address of a single LLC on the data link.

group (multi-cast) DSAP address. A destination address assigned to a collection of LLC's to facilitate their being addressed collectively. The least significant bit shall be set equal to "1."

higher layer. The conceptual layer of control or processing logic existing in the hierarchical structure of a station that is above the data link layer and upon which the performance of data link layer functions are dependent; for example, device control, buffer allocation, LLC station management, etc.

information field (I). The sequence of octets occurring between the control field and the end of the LLC PDU. The information field contents of I, TEST and UI PDU's are not interpreted at the LLC sublayer.

invalid frame. A PDU that either: a. Does not contain an integral number of octets, b. Does not contain at least two address octets and a control octet, c. Is identified by the physical layer or MAC sublayer as containing data bit errors.

LLC. That part of a data station that supports the logical link control functions of one or more logical links. The LLC generates command PDU's and response PDU's for transmission, and interprets received command PDU's and response

PDUs. Specific responsibilities assigned to a LLC include:

(1) Initiation of control signal interchange

(2) Organization of data flow

(3) Interpretation of received command PDU's and generation of appropriate response PDU's

(4) Actions regarding error control and error recovery functions in the LLC Sublayer.

MAC. That part of a data station that supports the medium access control functions that reside just below the Logical Link Control sublayer. The MAC procedures include framing/deframing data units, performing error checking, and acquiring the right to use the underlying physical medium.

N-layer. A subdivision of the architecture, constituted by subsystems of the same rank (N).

N-user. An N+1 entity that uses the services of the N-layer, and below, to communicate with another N+1 entity.

octet. A bit-oriented element that consists of eight contiguous binary bits.

peer protocol. The sequence of message exchanges between two entities in the same layer that utilize the services of the underlying layers to effect the successful transfer of data and/or control information from one location to another location.

protocol data unit (PDU). The sequence of contiguous octets delivered as a unit from or to the MAC Sublayer. A valid LLC PDU is at least 3 octets in length, and contains two address fields and a control field. A PDU may or may not include an information field in addition.

response. In data communications, a reply represented in the control field of a response PDU. It advises the addressed destination LLC of the action taken by the source LLC to one or more command PDU's.

response PDU. All PDU's sent by a LLC in which the C/R bit is equal to "1."

services. The capabilities and features provided by an N-layer to an N-user.

service class (use in primitives). A parameter used to convey the type of service required or desired (eg, priority).

2. LLC Sublayer Interface Service Specifications

This section covers the services required of, or by, the Logical Link Control (LLC) Sublayer at the logical interfaces with the Network Layer, the MAC Sublayer, and the LLC Sublayer Management function.

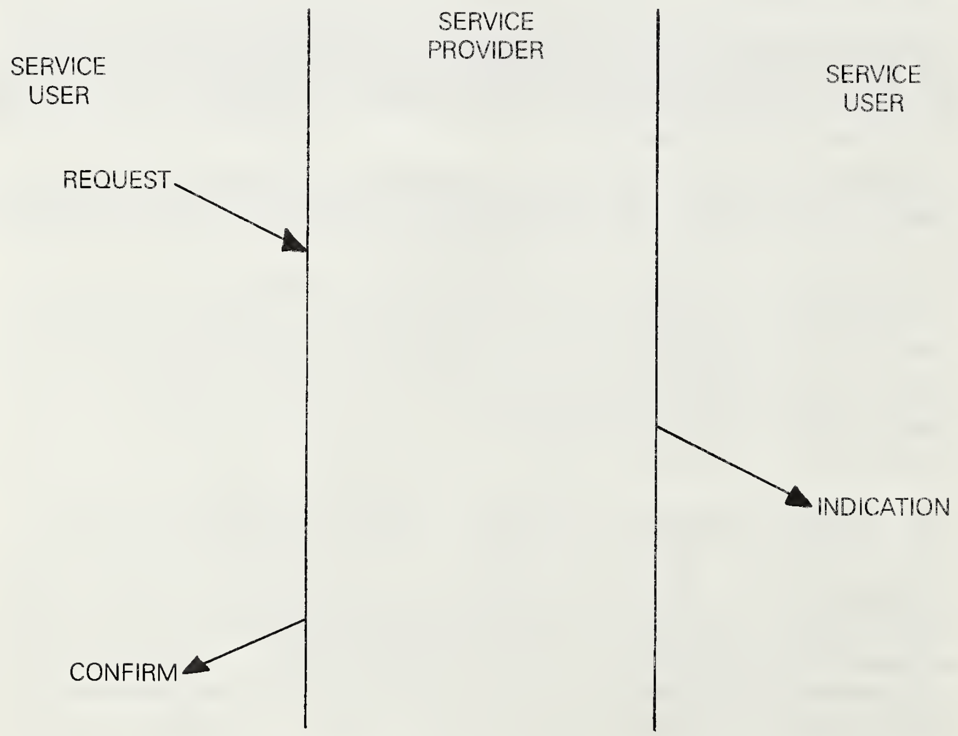
In general, the services of a layer (or sublayer) are the capabilities which it offers to a user in the next higher layer (or sublayer). In order to provide its

service, a layer (or sublayer) builds its functions on the services which it requires from the next lower layer (or sublayer). Fig 2-1 illustrates this notion of service hierarchy and shows the relationship of the two correspondent n-users and their associated n-layer (or sublayer) peer protocol entities.

Services are specified by describing the information flow at the interface between the n-user and the n-layer (or sublayer). This information flow is modeled by discrete, instantaneous interface events, which characterize the provision of a service. Each event consists of passing a service primitive from one layer (or sublayer) to the other through an n-layer (or sublayer) service access point associated with an n-user. Service primitives convey the information required in providing a particular service. These service primitives are an abstraction in that they specify only the service provided rather than the means by which the service is provided. This definition of service is independent of any particular interface implementation.

Services are specified by describing the service primitives and parameters which characterize each service. A service may have one or more related primitives which constitute the interface activity which is related to the particular service. Each service primitive may have zero or more parameters which convey the information required to provide the service.

Fig 2-1
Service Primitives



Primitives are of three generic types:

REQUEST The request primitive is passed from the n-user to the n-layer (or sublayer) to request that a service be initiated.

INDICATION The indication primitive is passed from the n-layer (or sublayer) to the n-user to indicate an internal n-layer (or sublayer) event which is significant to the n-user. This event may be logically related to a remote service request, or may be caused by an event internal to the n-layer (or sublayer).

CONFIRM The confirm primitive is passed from the n-layer (or sublayer) to the n-user to convey the results of one or more associated previous service request(s). This primitive may indicate either failure to comply or some level of compliance. It does not necessarily indicate any activity at the remote peer interface.

Possible relationships among primitive types are illustrated by the time sequence diagrams shown in Fig 2-2. The figure also indicates the logical relationship of the primitive types. Primitive types which occur earlier in time and are connected by dotted lines in the diagrams are the logical antecedents of subsequent primitive types. Note that the logical and time relationship of the indication and the confirm primitive types are specified by the semantics of a particular service.

2.1 Network Layer/LLC Sublayer Interface Service Specification. This section specifies the services required of the Logical Link Control (LLC) Sublayer by the Network Layer, as viewed from the Network Layer, to allow a local Network Layer entity to exchange packets with remote peer Network Layer entities. The services are described in an abstract way and do not imply any particular implementation or any exposed interface.

Two forms of service are provided—*unacknowledged connectionless* and *connection-oriented:*

Unacknowledged Connectionless Service. The data transfer service that provides the means by which network entities can exchange link service data units (LSDU's) without the establishment of a data link level connection. The data transfer can be point-to-point, multi-cast, or broadcast.

Connection-Oriented Services. This set of services provides the means for establishing, using, resetting, and terminating data link layer connections. These connections are point-to-point connections between LSAP's.

The connections establishment service provides the means by which a network entity can request, or be notified of, the establishment of data link layer connections.

The connection-oriented data transfer service provides the means by which a network entity can send or receive LSDU's over a data link layer connection. This service also provides data link layer sequencing, flow control, and error recovery.

The connection reset service provides the means by which established connections can be returned to the initial state.

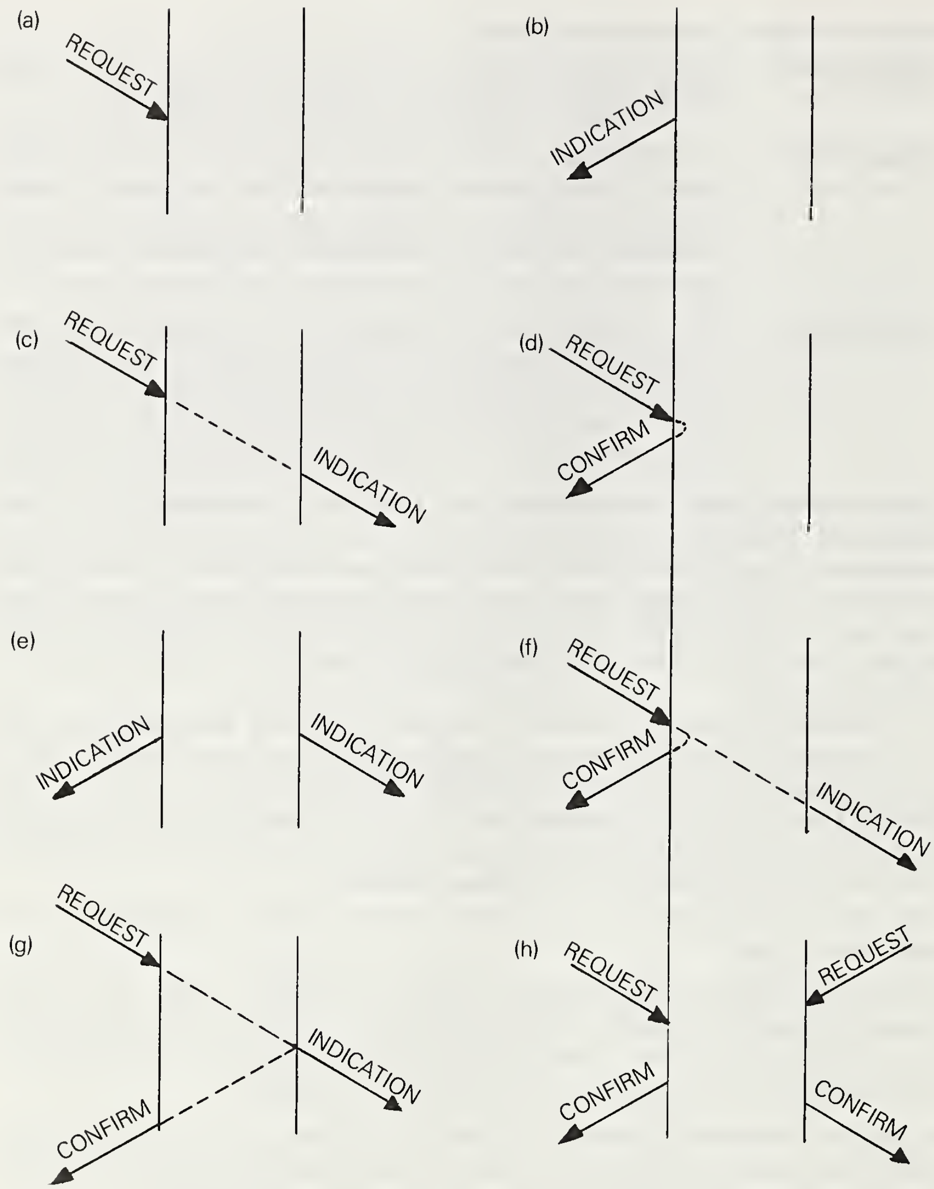


Fig 2-2
Time Sequence Diagram

The connection termination service provides the means by which a network entity can request or be notified of the termination of data link layer connections.

The connection flow control service provides the means to control the flow of data associated with a specified connection, across the network layer/data link layer interface.

2.1.1 Overview of Interactions.

2.1.1.1 Unacknowledged Connectionless Services.

2.1.1.1.1 Unacknowledged Connectionless Data Transfer. The primitives associated with unacknowledged connectionless data transfer are:

L_DATA.request

L_DATA.indication

The L_DATA.request primitive is passed to the LLC sublayer to request that an LSDU be sent using unacknowledged connectionless procedures. The L_DATA.indication primitive is passed from the LLC sublayer to indicate the arrival of an LSDU.

2.1.1.2 Connection-Oriented Services.

2.1.1.2.1 Connection Establishment. The service primitives associated with connection establishment are:

L_CONNECT.request

L_CONNECT.indication

L_CONNECT.confirm

The L_CONNECT.request primitive is passed to the LLC sublayer to request that a logical link connection be established between a local LSAP and a remote LSAP. The L_CONNECT.indication primitive is passed from the LLC sublayer to indicate the results of an attempt by a remote entity to establish a connection to a local LSAP. The L_CONNECT.confirm primitive is passed from the LLC sublayer to convey the results of the previous associated L_CONNECT.request primitive.

2.1.1.2.2 Connection-Oriented Data Transfer. The primitives associated with connection-oriented data transfer are:

L_DATA_CONNECT.request

L_DATA_CONNECT.indication

L_DATA_CONNECT.confirm

The L_DATA_CONNECT.request primitive is passed to the LLC sublayer to request that an LSDU be sent using connection-oriented procedures. The L_DATA_CONNECT.indication primitive is passed from the LLC sublayer to indicate the arrival of an LSDU. The L_DATA_CONNECT.confirm primitive is passed by the LLC sublayer to convey the results of the previously associated L_DATA_CONNECT.request primitive.

2.1.1.2.3 Connection Termination. The primitives associated with connection termination are:

L_DISCONNECT.request

L_DISCONNECT.indication

L_DISCONNECT.confirm

The L_DISCONNECT.request primitive is passed to the LLC sublayer to request the immediate termination of a link connection. The L_DISCONNECT.indication primitive is passed from the LLC sublayer to indicate to the network layer that a connection has been terminated. The L_DISCONNECT.confirm primitive is passed by the LLC sublayer to convey the results of the previous associated L_DISCONNECT.request primitive.

2.1.1.2.4 Connection Reset. The primitives associated with connection resetting are:

L_RESET.request

L_RESET.indication

L_RESET.confirm

The L_RESET.request primitive is passed to the LLC sublayer to request that a connection be immediately reset to the initial state. The L_RESET.indication primitive is passed from the LLC sublayer to indicate that a connection has been reset. The L_RESET.confirm primitive is passed from the LLC sublayer to convey the results of the previous associated L_RESET.request primitive.

2.1.1.2.5 Connection Flow Control. The primitives associated with connection flow control are:

L_CONNECTION_FLOWCONTROL.request

L_CONNECTION_FLOWCONTROL.indication

The L_CONNECTION_FLOWCONTROL.request primitive is passed to the LLC sublayer to control the flow from the LLC sublayer of L_DATA_CONNECT.indication primitives related to a connection. The L_CONNECTION_FLOWCONTROL.indication primitive is passed from the LLC sublayer to control the flow from the network layer of L_DATA_CONNECT.request primitives related to a connection.

2.1.2. Detailed Service Specifications. This section describes in detail the primitives and parameters associated with the identified services. Note that the parameters are specified in an abstract sense. The parameters specify the information that must be available to the receiving entity. A specific implementation is not constrained in the method of making this information available. The LLC sublayer may provide local confirm mechanisms for request type primitives.

The "local_address" and "remote_address" parameters provide at a minimum the logical concatenation of the MAC address field (SA and/or DA) and the LLC address field (SSAP and/or DSAP). An implementation of connection-oriented services may make use of a locally significant connection identifier to imply local and remote address parameters. The "l_sdu" parameter may be provided by actually passing the link service data unit, by passing a pointer, or by other means. The "service_class" parameter provides the priority associated with the data unit transfer. The "service_class" parameter is passed transparently to the underlying MAC sublayer via the appropriate LLC/MAC primitives, see 2.2. The "status" parameter indicates the degree of success associated with the data unit transfer. The "reason" parameter provides an explanation of the disconnection,

including a request by the remote entity, or an error internal to the LLC sublayer. The "amount" parameter provides information regarding the amount of data that the LLC entity is allowed to pass.

2.1.2.1 L_DATA.request.

2.1.2.1.1 Function. This primitive is the service request primitive for the unacknowledged connectionless data transfer service.

2.1.2.1.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DATA.request(
                local_address,
                remote_address,
                l_sdu,
                service_class
                )
```

The local_address and remote_address parameters specify the local and remote LSAP's involved in the data unit transfer. The remote_address may specify either an individual or group address. The l_sdu parameter specifies the link service data unit to be transferred by the link layer entity. The service_class parameter specifies the priority desired for the data unit transfer.

2.1.2.1.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request that a LSDU be sent to one or more remote LSAP(s) using unacknowledged connectionless procedures.

2.1.2.1.4 Effect on Receipt. Receipt of this primitive causes the LLC sublayer to attempt to send the LSDU using unacknowledged connectionless procedures.

2.1.2.1.5 Additional Comments. This primitive is independent of any connection with the remote LSAP.

A possible logical sequence of primitives associated with successful unacknowledged connectionless data unit transfer is illustrated in Fig 2-2 (c).

2.1.2.2 L_DATA.indication.

2.1.2.2.1 Function. This primitive is the service indication primitive for the unacknowledged connectionless data unit transfer service.

2.1.2.2.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DATA.indication(
                   local_address,
                   remote_address,
                   l_sdu,
                   service_class
                   )
```

The local_address and remote_address parameters specify the local and remote LSAP's involved in the data unit transfer. The local address may be the address of a local LSAP, or may be a group address specifying multiple LSAP's, including a local LSAP. The l_sdu parameter specifies the link service data unit

which has been received by the LLC sublayer entity. The service_class parameter specifies the priority desired for the data unit transfer.

2.1.2.2.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate the arrival of an LSDU from the specified remote entity.

2.1.2.2.4 Effect on Receipt. The effect of receipt of this primitive by the network layer is unspecified.

2.1.2.2.5 Additional Comments. This primitive is independent of any connection with the remote LSAP.

In the absence of errors, the contents of the l_sdu parameter are logically complete and unchanged relative to the l_sdu parameter in the associated L_DATA.request primitive.

2.1.2.3 L_CONNECT.request.

2.1.2.3.1 Function. This primitive is the service request primitive for the connection establishment service.

2.1.2.3.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_CONNECT.request(
                local_address,
                remote_address,
                service_class
                )
```

The local_address and remote_address parameters specify the local and remote LSAP's which are to be connected. The service_class parameter specifies the priority desired for the connection.

2.1.2.3.3 When Generated. This primitive is passed from the network layer to the LLC sublayer when the network layer entity wishes to establish a logical link connection, of a given service class, to a remote LSAP.

2.1.2.3.4 Effect on Receipt. The receipt of this primitive by the LLC sublayer causes the local LLC entity to initiate the establishment of a connection with the remote LLC entity.

2.1.2.3.5 Additional Comments. A possible logical sequence of primitives associated with successful connection establishment is illustrated in Fig 2-2 (g).

2.1.2.4 L_CONNECT.indication.

2.1.2.4.1 Function. This primitive is the service indication primitive for the connection establishment service.

2.1.2.4.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_CONNECT.indication(
                local_address,
                remote_address,
                status,
                service_class
                )
```

The local_address and remote_address parameters specify the local and remote LSAP's which are to be connected. One status code must indicate a successful connection attempt. Other status codes indicate the reason for failure. The service_class parameter indicates the priority desired for the connection, if the attempt was successful.

2.1.2.4.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate either that a connection of a certain service class has been established, or that a connection attempt has been refused.

2.1.2.4.4 Effect on Receipt. The network entity may use this connection for data unit transfer.

2.1.2.4.5 Additional Comments. The basis for connection refusal by the LLC sublayer is implementation dependent but may include such reasons as access permission, or unacceptable connection service class.

If the network layer entity does not wish to maintain a connection, then a L_DISCONNECT.request primitive is issued.

2.1.2.5 L_CONNECT.confirm.

2.1.2.5.1 Function. This primitive is the service confirm primitive for the connection establishment service.

2.1.2.5.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_CONNECT.confirm(
                    local_address,
                    remote_address,
                    status,
                    service_class
                    )
```

The local_address and remote_address parameters specify the local and remote LSAP's which are to be connected. One status code must indicate a successful connection attempt. Other status codes indicate the reason for failure. The service_class parameter indicates the priority provided for the connection, if the attempt was successful.

2.1.2.5.3 When Generated. This primitive is passed by the LLC sublayer to the network layer to convey the results of the previous associated L_CONNECT.request primitive. The results indicate either that the connection attempt was successful, or that a connection of the specified service class could not be achieved.

2.1.2.5.4 Effect on Receipt. The network entity may use this connection for data unit transfer.

2.1.2.5.5 Additional Comments. If the connection attempt was successful, this primitive indicates that the remote LLC sublayer entity received and positively acknowledged the connection attempt, and that it passed an L_CONNECT.indication to the remote user network entity.

2.1.2.6 L_DATA_CONNECT.request.

2.1.2.6.1 Function. This primitive is the service request primitive for the connection-oriented data unit transfer service.

2.1.2.6.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DATA_CONNECT.request(
                        local_address,
                        remote_address,
                        l_sdu
                        )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection. The l_sdu parameter specifies the link service data unit to be transferred by the LLC sublayer entity.

2.1.2.6.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request that a LSDU be transferred to a remote LSAP over an existing connection.

2.1.2.6.4 Effect on Receipt. The receipt of this primitive by the LLC sublayer causes the LLC sublayer to transfer the LSDU over the specified connection using connection-oriented procedures.

2.1.2.6.5 Additional Comments. A possible logical sequence of primitives associated with successful connection-oriented data unit transfer is illustrated in Fig 2-2(g).

2.1.2.7 L_DATA_CONNECT.indication.

2.1.2.7.1 Function. This primitive is the service indication primitive for the connection-oriented data unit transfer service.

2.1.2.7.2 Semantics of the Service Primitive. The primitive shall provide parameters, as follows:

```
L_DATA_CONNECT.indication(
                           local_address,
                           remote_address,
                           l_sdu
                           )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection. The l_sdu parameter specifies the link service data unit which has been received by the LLC sublayer entity.

2.1.2.7.3 When Generated. This primitive is passed by the LLC sublayer to the network layer to indicate the arrival of an LSDU from the specified remote network layer entity over a particular connection.

2.1.2.7.4 Effect on Receipt. The effect of receipt of this primitive by the network layer is unspecified.

2.1.2.7.5 Additional Comments. The L_DATA_CONNECT.request primitive does not contain a service_class parameter because service class must be uniform for all L_DATA_CONNECT.requests in a particular connection.

In the absence of errors, the contents of the l_sdu parameter are logically complete and unchanged relative to the l_sdu parameter in the associated L_DATA.request primitive.

2.1.2.8 L_DATA_CONNECT.confirm.

2.1.2.8.1 Function. This primitive is the service confirm primitive for the connection-oriented data unit transfer service.

2.1.2.8.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DATA_CONNECT.confirm(
                        local_address,
                        remote_address,
                        status
                        )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection. The status parameter indicates the success or failure of the one or more previous associated connection-oriented data unit transfer request(s).

2.1.2.8.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate the success or failure of the one or more previous associated connection-oriented data unit transfer request(s).

2.1.2.8.4 Effect on Receipt. The effect of receipt of this primitive by the network layer is unspecified.

2.1.2.8.5 Additional Comments. If the transfer was successful this primitive indicates that the remote LLC sublayer entity received and positively acknowledged the LSDU.

2.1.2.9 L_DISCONNECT.request.

2.1.2.9.1 Function. This primitive is the service request primitive for the connection termination service.

2.1.2.9.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DISCONNECT.request(
                      local_address,
                      remote_address,
                      )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection to be terminated.

2.1.2.9.3 When Generated. This primitive is passed from the network layer to the LLC sublayer when the network layer entity wishes to terminate a connection.

2.1.2.9.4 Effect on Receipt. Receipt of this primitive causes the LLC sublayer to immediately terminate the connection.

2.1.2.9.5 Additional Comments. All unacknowledged LSDU's are discarded. The connection termination service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful disconnection (ie, without loss of data) is the responsibility of a higher layer protocol.

A possible logical sequence of primitives associated with successful connection termination is illustrated in Fig 2-2(g).

2.1.2.10 L_DISCONNECT.indication.

2.1.2.10.1 Function. This primitive is the service indication primitive for the connection termination service.

2.1.2.10.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DISCONNECT.indication(
                        local_address,
                        remote_address,
                        reason
                        )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the terminated connection. The reason parameter specifies the reason for the disconnection. The reasons for disconnection may include a request by the remote entity, or an error internal to the LLC sublayer.

2.1.2.10.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to inform the network layer that a connection has been terminated.

2.1.2.10.4 Effect on Receipt. The network entity may no longer use this connection for data unit transfer.

2.1.2.10.5 Additional Comments. All unacknowledged LSDU's are discarded. The connection termination service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful disconnection (ie, without loss of data) is the responsibility of a higher layer protocol.

2.1.2.11 L_DISCONNECT.confirm.

2.1.2.11.1 Function. This primitive is the service confirm primitive for the connection termination service.

2.1.2.11.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_DISCONNECT._confirm(
                        local_address,
                        remote_address,
                        status
                        )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the terminated connection. The status parameter indicates whether or not the remote LLC sublayer entity acknowledged the disconnection attempt.

2.1.2.11.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to inform the network layer entity when connection termination is complete.

2.1.2.11.4 Effect on Receipt. The network entity may no longer use this connection for data unit transfer.

2.1.2.11.5 Additional Comments. The L_DISCONNECT.confirm primi-

tive indicates that the remote LLC sublayer has acknowledged the disconnect. The remote LLC sublayer entity is expected to pass an L_DISCONNECT.indication to the remote user network layer.

2.1.2.12 L_RESET.request.

2.1.2.12.1 Function. This primitive is the service request primitive for the connection reset service.

2.1.2.12.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_RESET.request(
                    local_address,
                    remote_address
                    )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection to be reset.

2.1.2.12.3 When Generated. This primitive is passed from the network layer to the LLC sublayer to request that a connection be reset to the initial state.

2.1.2.12.4 Effect on Receipt. Receipt of this primitive causes immediate resetting of the connection.

2.1.2.12.5 Additional Comments. All unacknowledged LSDU's are discarded. The connection reset service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful reset (ie, without loss of data) is the responsibility of a higher layer protocol.

A possible logical sequence of primitives associated with successful connection reset is illustrated in Fig 2-2(g).

2.1.2.13 L_RESET.indication.

2.1.2.13.1 Function. This primitive is the service indication primitive for the connection reset service.

2.1.2.13.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_RESET.indication(
                    local_address,
                    remote_address,
                    reason
                    )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the reset connection. The reason parameter specifies the reason for the connection reset. One of the reason codes indicates that the reset was requested by a remote network entity. All other codes indicate that the reset originated within the LLC sublayer, (as shown in Fig 2-2(e)) and may optionally provide implementation dependent information about the cause of the reset.

2.1.2.13.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to indicate that the connection has been reset.

2.1.2.13.4 Effect on Receipt. The effect of receipt of this primitive by the network layer is unspecified.

2.1.2.13.5 Additional Comments. The reasons for the reset may include a request by the remote entity or an error internal to the LLC sublayer. All unacknowledged LSDU's are discarded. The connection reset service is an abortive service. That is, no guarantee of delivery can be assumed about data that is not yet acknowledged at a higher layer. Thus, graceful reset (ie, without loss of data) is the responsibility of a higher layer protocol.

2.1.2.14 L_RESET.confirm.

2.1.2.14.1 Function. This primitive is the service confirm primitive for the connection reset service.

2.1.2.14.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_RESET.confirm(
                    local_address,
                    remote_address,
                    status
                    )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the reset connection. The reason parameter indicates whether or not the remote LLC sublayer entity acknowledged the reset attempt.

2.1.2.14.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to inform the network layer that a connection reset has been completed.

2.1.2.14.4 Effect on Receipt. The network entity may use this connection for data unit transfer.

2.1.2.14.5 Additional Comments. This primitive indicates that the remote LLC sublayer entity has acknowledged the reset. The remote LLC sublayer entity is expected to pass an L_RESET.indication to the remote user network entity.

2.1.2.15 L_CONNECTION_FLOWCONTROL.request.

2.1.2.15.1 Function. This primitive is the service request primitive for the connection flow control service.

2.1.2.15.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_CONNECTION_FLOWCONTROL.request(
                                    local_address,
                                    remote_address,
                                    amount
                                    )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection to be flow controlled. The amount parameter specifies the amount of data the LLC sublayer entity is permitted to pass.

2.1.2.15.3 When Generated. This primitive is passed from the network

layer to the LLC sublayer to request control of the flow of L_DATA_CONNECT.indication primitives associated with a connection from the LLC sublayer.

2.1.2.15.4 Effect on Receipt. Receipt of this primitive causes the LLC sublayer to adjust the amount of data that may be passed to the network layer.

2.1.2.15.5 Additional Comments. Control of the flow of data on a connection is independent of control of the flow on other connections. The amount of data permitted to be passed is dynamically updated by each request. If amount is specified as zero, then the associated flow is stopped. Specific implementations may allow amount to be specified in implementation specific units, and may allow amount to be specified as "infinite."

A possible logical sequence of primitives associated with an L_CONNECTION_FLOWCONTROL.request is illustrated in Fig 2-2 (a).

2.1.2.16 L_CONNECTION_FLOWCONTROL.indication.

2.1.2.16.1 Function. This primitive is the service indication primitive for the connection flow control service.

2.1.2.16.2 Semantics of the Service Primitive. The primitive shall provide parameters as follows:

```
L_CONNECTION_FLOWCONTROL.indication(
                                    local_address,
                                    remote_address,
                                    amount
                                    )
```

The local_address and remote_address parameters specify the local and remote LSAP's of the connection to be controlled. The amount parameter specifies the amount of data which the network entity is permitted to pass to avoid data loss.

2.1.2.16.3 When Generated. This primitive is passed from the LLC sublayer to the network layer to request control of the flow of L_DATA_CONNECT.request primitives associated with a connection from the network layer.

2.1.2.16.4 Effect on Receipt. Receipt of the primitive causes the network layer to adjust the amount of data that it is allowed to pass without data loss.

2.1.2.16.5 Additional Comments. Control of the flow of data on a connection is independent of control of the flow on other connections. The amount of data permitted to be passed is dynamically updated by each request. If amount is specified as zero, then the associated flow is stopped. Specific implementations may allow amount to be specified in implementation specific units, and may allow amount to be specified as "infinite."

A possible logical sequence of primitives associated with an L_CONNECTION_FLOWCONTROL.indication is illustrated in Figure 2-2 (b).

2.2 LLC Sublayer/MAC Sublayer Interface Service Specification. This section specifies the services required of the Medium Access Control (MAC)

sublayer by the Logical Link Control (LLC) sublayer to allow the local LLC sublayer entity to exchange LLC data units with peer LLC sublayer entities. The services are described in an abstract way and do not imply any particular implementation or any exposed interface.

2.2.1 Overview of Interactions.

- MA_DATA.request
- MA_DATA.indication
- MA_DATA.confirm

2.2.2 Detailed Service Specification.

2.2.2.1 MA_DATA.request.

2.2.2.1.1 Function. This primitive defines the transfer of a MSDU from a local LLC sublayer entity to a single peer LLC entity, or multiple peer LLC entities in the case of group addresses.

2.2.2.1.2 Semantics of the Service Primitive. The semantics of the primitive are as follows:

```
MA_DATA.request(
                    destination_address,
                    m_sdu,
                    requested_service_class
                    )
```

The destination_address parameter shall specify either an individual or a group MAC entity address. It must contain sufficient information to create the DA field that is appended to the frame by the local MAC sublayer entity as well as any Physical Layer address information (eg, transmit frequency in broadband applications). The m_sdu parameter specifies the MAC service data unit to be transmitted by the MAC sublayer entity, which includes the DSAP, SSAP, C, and information (if present) fields as specified in Section 3, as well as sufficient information for the MAC sublayer entity to determine the length of the data unit. The requested_service_class parameter specifies the priority desired for the data unit transfer.

2.2.2.1.3 When Generated. This primitive is generated by the LLC sublayer entity whenever a MSDU must be transferred to a peer LLC entity or entities. This can be as a result of a request from higher layers or protocol, or from a MSDU generated internally to the LLC sublayer, such as required by Type 2 operation.

2.2.2.1.4 Effect of Receipt. The receipt of this primitive shall cause the MAC entity to append all MAC specified fields, including DA, SA, and any fields that are unique to the particular medium access method, and pass the properly formatted frame to the lower layers of protocol for transfer to the peer MAC sublayer entity or entities.

2.2.2.1.5 Additional Comments. None.

2.2.2.2. MA_DATA.indication.

2.2.2.2.1 Function. This primitive defines the transfer of a MSDU from

the MAC sublayer entity to the LLC sublayer entity, or entities in the case of group addresses. In the absence of errors, the contents of the m_sdu parameter are logically complete and unchanged relative to the m_sdu parameter in the associated MA_DATA.request primitive.

2.2.2.2.2 Semantics of the Service Primitive. The semantics of the primitive are as follows:

```
MA_DATA.indication(
                    destination_address,
                    source_address,
                    m_sdu,
                    reception_status,
                    requested_service_class
                    )
```

The destination_address parameter shall be either an individual or a group address as specified by the DA field of the incoming frame. The source_address parameter must be an individual address as specified by the SA field of the incoming frame. The m_sdu parameter specifies the MAC service data unit as received by the local MAC entity. The reception_status parameter indicates the success or failure of the incoming frame. The requested_service_class parameter specifies the service class desired for this data unit transfer.

2.2.2.2.3 When Generated. The MA_DATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities to indicate the arrival of a frame at the local MAC sublayer entity. Frames are reported only if at the MAC sublayer they are validly formated, received without error, and their destination address designates the local MAC entity.

2.2.2.2.4 Effect of Receipt. The effect of receipt of this primitive by the LLC sublayer is dependent on the validity and content of the frame.

2.2.2.2.5 Additional Comments. If the local MAC sublayer entity is designated by the destination_address parameter of an MA_DATA.request primitive, the indication primitive will also be invoked by the MAC entity to the local LLC entity. This full duplex characteristic of the MAC sublayer may be due to unique functionality within the MAC sublayer or full duplex characteristics of the lower layers, (eg, all frames transmitted to the broadcast address will invoke MA_DATA.indication primitives at all stations in the network including the station that generated the request.)

2.2.2.3 MA_DATA.confirm.

2.2.2.3.1 Function. This primitive has local significance and shall provide an appropriate reply to the LLC sublayer MA_DATA.request primitive signifying the success or failure of the request.

2.2.2.3.2 Semantics of the Service Primitive. The semantics of this primitive are as follows:

```
MA_DATA.confirm(
                transmission_status,
                provided_service_class
                )
```

The transmission_status parameter is used to pass status information back to the local requesting LLC sublayer entity. It is used to indicate the success or failure of the previous associated MA_DATA.request primitive. The types of failures that can be associated with this primitive are dependent on the particular implementation as well as the type of Medium Access Control sublayer that is used (eg, excessive collisions may be a failure returned by a CSMA/CD MAC sublayer entity). The provided_service_class parameter specifies the service class that was provided for the data unit transfer.

2.2.2.3.3. When Generated. This primitive is generated in reply to an MA_DATA.request primitive from the local LLC sublayer entity.

2.2.2.3.4 Effect of Receipt. The effect of receipt of this primitive by the LLC sublayer is unspecified.

2.2.2.3.5 Additional Comments. It is assumed that sufficient information is available to the LLC sublayer to associate the confirm with the appropriate request.

2.3 LLC Sublayer/LLC Sublayer Management Function Interface Service Specification. (This matter is the subject of further ongoing study and resolution.)

3. LLC Protocol Data Unit (PDU) Structure

3.1 General. This section defines in detail the Logical Link Control (LLC) Protocol Data Unit (PDU) structure for data communication systems using bit-oriented procedures. It defines the relative positions of the various components of the Protocol Data Unit (PDU). It defines the method for representing Data Link Layer service access point addresses (to or from Network Layer entities). It defines a partition of these addresses into individual and group addresses. Details of the control and information field allocation are specified in Section 5.

3.2 LLC PDU Format. All LLC PDU's shall conform to the format shown in Fig 3-1:

Fig 3-1
LLC PDU Format

DSAP Address	SSAP Address	Control	Information
8 bits	8 bits	y bits	8*M*bits

where:

DSAP address = destination service access point address field,
SSAP address = source service access point address field,
Control = control field,
16 bits for formats that include sequence numbering, and 8 bits for formats that do not (see 5.2),
Information = information field,
* = multiplication, and
M = an integer value equal to or greater than 0. (Upper bound of M is a function of the medium access control methodology used.)

3.3 Elements of the LLC PDU.

3.3.1 Address Fields. *Each LLC PDU shall contain two address fields: the Destination Service Access Point (DSAP) Address field and the Source Service Access Point (SSAP) Address field, in that order. Each address field shall contain only a single address. The DSAP Address field shall identify the one, or more, service access points for which the LLC information field is intended. The SSAP Address field shall identify the specific service access point from which the LLC information field was initiated.

3.3.1.1 Address Representation. The representation of each address field shall be as shown in Fig 3-2a and 3-2b:

(1) Each address field shall contain one octet.

(2) Each address field shall contain 7 bits of actual address and one bit that shall be used in the DSAP Address field to identify the DSAP Address as either an individual or a group address (called the address type designation bit) and in the SSAP Address field to identify that the LLC PDU is a command or a response (called the command/response identifier bit).

(3) The address type designation bit shall be located in the least significant bit position of the DSAP address field. If this bit is "0," it shall indicate that the address is an individual DSAP address. If this bit is "1," it shall indicate that the address is a group DSAP address that identifies none, one or more, or all of the service access points that are serviced by the LLC entity.

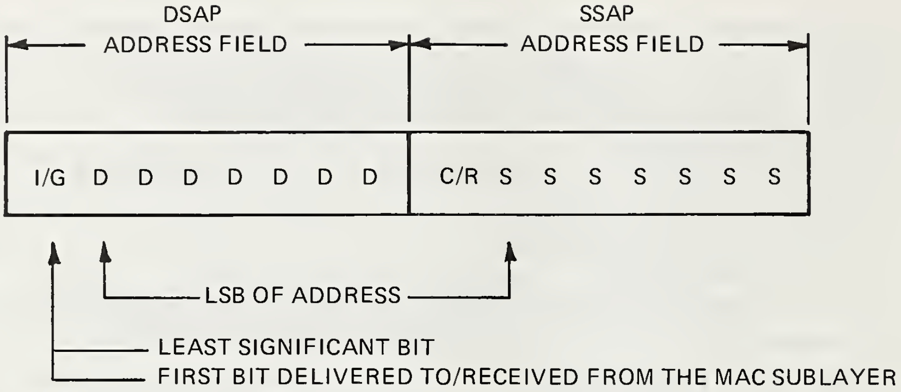
(4) The command/response identifier bit shall be located in the least significant bit position of the SSAP address field. If this bit is "0," it shall indicate that the LLC PDU is a command. If this bit is "1," it shall indicate that the LLC PDU is a response.

3.3.1.2 Address Usage. An individual address shall be usable as both a SSAP and a DSAP address; a null address shall be usable as both an SSAP and a DSAP address; a group address shall be usable only as a DSAP address.

All "1"s in the DSAP address field (ie, the address type designation bit set to "1," and the seven address bits set to "1") is predefined to be the "Global" DSAP address. This DSAP address designates a group consisting of all DSAP's actively being serviced by the underlying MAC Service Access Point Address(es).

All "0"s in the DSAP or SSAP address field, (ie, the address type designation bit set to "0," and the seven address bits set to "0") is predefined to be the "Null"

*A number of specific LSAP code points have been identified for particular uses. A list of these code points can be obtained from the IEEE Standards Office.



I/G = 0 INDIVIDUAL DSAP
I/G = 1 GROUP DSAP
C/R = 0 COMMAND
C/R = 1 RESPONSE

XODDDDDD DSAP ADDRESS
XOSSSSSS SSAP ADDRESS

X1DDDDDD
X1SSSSSS RESERVED FOR IEEE 802 DEFINITION

Fig 3-2a
DSAP and SSAP Address Field Formats

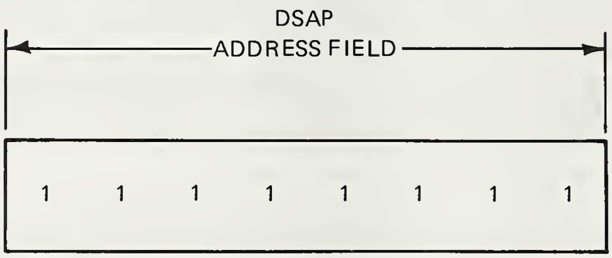


Fig 3-2b
Global DSAP Address Field Format

address. The Null service access point address designates the LLC that is associated with the underlying MAC Service Access Point address and is NOT used to identify any service access point to the Network Layer or any service access point to an associated Layer Management function.

Addresses 01000000 and 11000000 are designated as the individual and the group addresses, respectively, for an LLC sublayer management function at the station. Other addresses with the next to low-order bit set to "1" are reserved for IEEE Std 802 definition.

3.3.2 Control Field. The control field shall consist of one or two octets which shall be used to designate command and response functions, and which shall contain sequence numbers when required. The content of this field shall be as described in Section 5.

3.3.3 Information Field. The information field shall consist of any integral number (including zero) of octets.

3.3.4 Bit Order. Addresses, commands and responses, and sequence numbers shall be delivered to/received from the MAC sublayer least significant bit first (ie, the first bit of a sequence number that is delivered/received shall have the weight 2**0). The information field shall be delivered to the MAC sublayer in the same bit order as received from the Network Layer. The information field shall be delivered to the Network Layer in the same bit order as received from the MAC sublayer.

3.3.5 Invalid LLC PDU. An invalid LLC PDU shall be defined as one which meets at least one of the following conditions:

(1) It is identified as such by the Physical Layer or the medium access control (MAC) sublayer.

(2) It is not an integral number of octets in length.

(3) It does not contain two properly formatted address fields, one control field, and optionally an information field in their proper order.

(4) Its length is less than 3 octets (one-octet control field) or 4 octets (two-octet control field).

Invalid LLC PDU's shall be ignored.

4. LLC Types and Classes of Procedure

4.1 General. LLC defines two types of operation for data communication between service access points.

(1) **Type 1 Operation.** With Type 1 operation, PDU's shall be exchanged between LLC's without the need for the establishment of a data link connection. In the LLC Sublayer these PDU's shall not be acknowledged, nor shall there be any flow control or error recovery in the Type 1 procedures.

(2) **Type 2 Operation.** With Type 2 operation, a data link connection shall be established between two LLC's prior to any exchange of information-bearing PDU's. The normal cycle of communication between two Type 2 LLC's on a data link connection shall consist of the transfer of PDU's containing information from the source LLC to the destination LLC, acknowledged by a PDU in the opposite direction.

With Type 2 operation, the control of traffic between the source LLC and the destination LLC shall be effected by means of a numbering scheme, which shall be cyclic within a modulus of 128 and measured in terms of PDU's. An independent numbering scheme shall be used for each source/destination LLC pair. Each such pairing shall be defined to be a logical point-to-point data link

connection between Data Link Layer service access points and shall take into account the DA and SA addressing that is part of the medium access control (MAC) sublayer. The acknowledgment function shall be accomplished by the destination LLC informing the source LLC of the next expected sequence number. This shall be done either in a separate PDU, not containing information, or within the control field of a PDU containing information.

LLC Type 2 procedures shall be applicable to balanced data link connections. A balanced data link connection shall involve two participating LLC's. For control purposes, each LLC shall assume responsibility for the organization of its data flow and for the link error recovery operations for the transmissions that it originates. Each LLC shall be capable of sending and receiving both command and response PDU's.

For the transfer of data between LLC's in Type 2 operation, Fig 4-1 depicts the data link control functions utilized. The data source in each LLC shall control the data sink in the other LLC by the use of command PDU's. The information shall flow from the data source to the data sink, and any acknowledgments shall always be sent in the opposite direction. The poll-type command PDU's shall be utilized by each LLC station to solicit specific acknowledgments and status responses from the other LLC.

NOTE: The need for a reliable transaction type service has been identified as a subject for urgent further study. The desired service is one that includes a basic acknowledgment function that will indicate that a sent PDU has been received and accepted by the peer LLC sublayer.

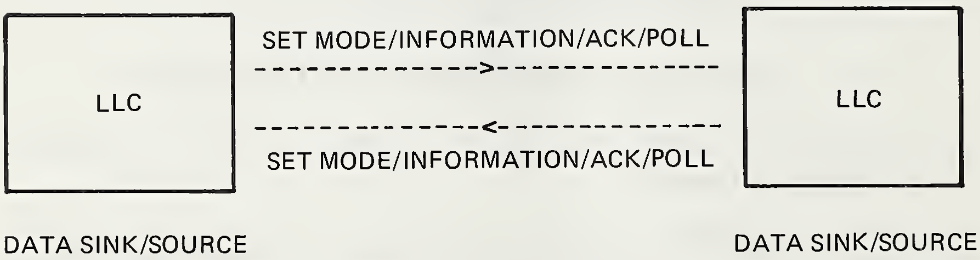


Fig 4-1
Balanced Data Link Connection Configuration

4.2 Classes of LLC's. Two classes of LLC's are defined. A Class I LLC shall support Type 1 operation only whereas a Class II LLC shall support both Type 1 and Type 2 operations (Fig 4-2).

This means all LLC's on a local area network shall have Type 1 operation in common. In a Class II LLC, the support of Type 1 operation shall be totally independent of the modes or change of modes of the Type 2 operation in that same LLC. A Class II LLC shall be capable of going back and forth between Type 1 operation and Type 2 operation on a PDU-to-PDU basis, if necessary.

TYPE OF OPERATION

CLASSES OF LLC	1	2
I	X	
II	X	X

Fig 4-2
Classes of LLC

4.2.1 Class I LLC. Class I LLC's shall support Type 1 operation only. Class I service shall be applicable to individual, group, global, and null DSAP addressing, and applications requiring no data link layer acknowledgment or flow control procedures. The set of command PDU's and response PDU's supported in Class I service shall be:

	Commands	Responses
Type 1:	UI	
	XID	XID
	TEST	TEST

4.2.2 Class II LLC. Class II LLC's shall support both Type 1 operation and Type 2 operation. In a Class II station, the operation of the Type 1 procedures and the Type 2 procedures are completely independent. The set of command PDU's and response PDU's supported in Class II service shall be:

	Commands	Responses
Type 1:	UI	
	XID	XID
	TEST	TEST
Type 2:	I	I
	RR	RR
	RNR	RNR
	REJ	REJ
	SABME	UA
	DISC	DM
		FRMR

5. LLC Elements of Procedure

5.1 General. This section specifies the elements of the local area network logical link control (LLC) procedures for code-independent data communication using the LLC PDU structure (see Section 3).

These LLC elements of procedure are defined specifically in terms of the actions that shall occur in the LLC on receipt of commands, and occasionally on receipt of a reply to a command, over a logical data link (Type 1) and a data link connection (Type 2). Each element of procedure is utilized by only one of the two types of operation (Type 1 or Type 2) which are defined in Section 4.

5.2 Control Field Formats. The three formats defined for the control field (Fig 5-2) shall be used to perform numbered information transfer, numbered supervisory transfer, unnumbered control, and unnumbered information transfer functions. The numbered information transfer and supervisory transfer functions apply only to Type 2 operation. The unnumbered control and unnumbered information transfer functions apply either to Type 1 or Type 2 operation (but not both) depending upon the specific function selected.

LLC PDU CONTROL FIELD BITS

	1	2	3	4	5	6	7	8	9	10 - 16
INFORMATION TRANSFER COMMAND/RESPONSE (I-FORMAT PDU)	0	N(S)							P/F	N(R)
SUPERVISORY COMMANDS/RESPONSES (S-FORMAT PDUs)	1	0	S	S	X	X	X	X	P/F	N(R)
UNNUMBERED COMMANDS/RESPONSE (U-FORMAT PDUs)	1	1	M	M	P/F	M	M	M		

Fig 5-1
LLC PDU Control Field Formats

where:

N(S)	=	Transmitter send sequence number (Bit 2 = low-order bit)
N(R)	=	Transmitter receive sequence number (Bit 10 = low-order bit)
S	=	Supervisory function bit
M	=	Modifier function bit
X	=	Reserved and set to zero
P/F	=	Poll bit - command LLC PDU transmissions Final bit - response LLC PDU transmissions (1 = Poll/Final)

5.2.1 Information Transfer Format I. The I-format PDU shall be used to perform a numbered information transfer in Type 2 operation. Except where otherwise specified (eg, UI, TEST, FRMR and XID), it shall be the only LLC PDU which may contain an information field. The functions of N(S), N(R), and P/F shall be independent; ie, each I-format PDU shall have an N(S) sequence number, an N(R) sequence number which shall or shall not acknowledge additional I-format PDU's at the receiving LLC, and a P/F bit that shall be set to "1" or "0."

5.2.2 Supervisory Format. S. The S-format PDU shall be used to perform data link supervisory, control functions in Type 2 operation such as acknowledging I-format PDU's, requesting retransmission of I-format, PDU's, and requesting a temporary suspension of transmission of I-format PDU's. The functions of N(R) and P/F shall be independent; ie, each S-format PDU shall have an N(R) sequence number which shall or shall not acknowledge additional I-format PDUs at the receiving LLC, and a P/F bit that shall be set to "1" or "0."

5.2.3 Unnumbered Format. U. The U-format PDU's shall be used in either Type 1 or Type 2 operation, depending upon the specific function utilized, to provide additional data link control functions and to provide unsequenced information transfer. The U-format PDU's shall contain no sequence numbers, but shall include a P/F bit that shall be set to "1" or "0."

5.3 Control Field Parameters. The various parameters associated with the control field formats are described in the following sections.

5.3.1 Type 1 Operation Parameters. The only parameter that exists in Type 1 operation is the Poll/Final (P/F) bit. The P/F bits set to "1" shall only be used in Type 1 operation with the XID and TEST command/response PDU functions. The Poll (P) bit set to "1" shall be used to solicit (poll) a correspondent response PDU with the F bit set to 1 from the addressed LLC. The Final (F) bit set to "1" shall be used to indicate that response PDU that is sent by the LLC as the result of a soliciting (poll) command PDU (P bit set to "1").

5.3.2 Type 2 Operation Parameters. The various parameters associated with the control field formats in Type 2 operation are described in the following sections.

5.3.2.1 Modulus. Each I PDU shall be sequentially numbered with a number which shall have a value between 0 and MODULUS minus ONE (where MODULUS is the modulus of the sequence numbers). The modulus shall equal 128 for the Type 2 LLC control field format. The sequence numbers shall cycle through the entire range.

The maximum number of sequentially numbered I PDU's that may be outstanding (ie, unacknowledged) in a given direction on a data link connection at any given time shall never exceed one less than the modulus of the sequence numbers. This restriction shall prevent any ambiguity in the association of sent I PDU's with sequence numbers during normal operation and/or error recovery action.

5.3.2.2 LLC PDU State Variables and Sequence Numbers. A station LLC shall maintain a send state variable V(S) for the I PDU's it sends and a

receive state variable V(R) for the I PDU's it receives on each data link connection. The operation of V(S) shall be independent of the operation of V(R).

5.3.2.2.1 Send State Variable V(S). The send state variable shall denote the sequence number of the next in-sequence I PDU to be sent on a specific data link connection. The send state variable shall take on a value between 0 and MODULUS minus ONE (where MODULUS equals 128 and the numbers cycle through the entire range). The value of the send state variable shall be incremented by one with each successive I PDU transmission on the associated data link connection, but shall not exceed N(R) of the last received PDU by more than MODULUS minus ONE.

5.3.2.2.2 Send Sequence Number N(S). Only I PDU's shall contain N(S), the send sequence number of the sent PDU. Prior to sending an in-sequence I PDU, the value of N(S) shall be set equal to the value of the send state variable for that data link connection.

5.3.2.2.3 Receive State Variable V(R). The receive state variable shall denote the sequence number of the next in-sequence I PDU to be received on a specific data link connection. The receive state variable shall take on a value between 0 and MODULUS minus ONE (where MODULUS equals 128 and the numbers cycle through the entire range). The value of the receive state variable associated with a specific data link connection shall be incremented by one whenever an error-free, in-sequence I PDU is received whose send sequence number N(S) equals the value of the receive state variable for the data link connection.

5.3.2.2.4 Receive Sequence Number N(R). All I-format PDU's and S-format PDU's shall contain N(R), the expected sequence number of the next received I PDU on the specified data link connection. Prior to sending an I-format PDU or S-format PDU, the value of N(R) shall be set equal to the current value of the associated receive state variable for that data link connection. N(R) shall indicate that the station sending the N(R) has received correctly all I PDU's numbered up through N(R)-1 on the specified data link connection.

5.3.2.3 Poll/Final (P/F) Bit. The poll (P) bit shall be used to solicit (poll) a response from the addressed LLC. The final (F) bit shall be used to indicate the response PDU sent as the result of a soliciting (poll) command.

The poll/final (P/F) bit shall serve a function in Type 2 operation in both command PDU's and response PDU's. In command PDU's the P/F bit shall be referred to as the P bit. In response PDU's it shall be referred to as the F bit. P/F bit exchange provides a distinct command/response linkage that is useful during both normal operation and recovery situations.

5.3.2.3.1 Poll Bit Functions. A command PDU with the P bit set to "1" shall be used on a data link connection to solicit a response PDU with the F bit set to "1" from the addressed LLC on that data link connection.

Only one PDU with a P bit set to "1" shall be outstanding in a given direction at a given time on the data link connection between any specific pair of LLC's. Before a LLC issues another PDU on the same data link connection with the P bit set to "1," the LLC shall have received a response PDU with the F bit set to "1" from the addressed LLC. If no valid response PDU is received within a system-

defined P-bit Timer time-out period, the (re)sending of a command PDU with the P bit set to "1" shall be permitted for error recovery purposes.

5.3.2.3.2 Final Bit Functions. A response PDU with the F bit set to "1" shall be used to acknowledge the receipt of a command PDU with the P bit set to "1."

Following the receipt of a command PDU with the P bit set to "1," the LLC shall send a response PDU with the F bit set to "1" on the appropriate data link connection at the earliest possible opportunity.

The LLC shall be permitted to send appropriate response PDU's with the F bit set to "0" at any medium access opportunity on an asynchronous basis (without the need for a command PDU).

5.4 Commands and Responses. This section defines the commands and associated responses. Sections 5.4.1 and 5.4.2 contain the definitions of the set of commands and responses (listed below) for each of the control field formats for Type 1 operation and for Type 2 operation, respectively.

The C/R bit, located in the low-order bit of the SSAP, is used to distinguish between commands and responses. The following discussion of commands and responses assumes that the C/R bit has been properly decoded.

Information transfer format commands	Information transfer format responses
I—Information	I—Information
Supervisory format commands	**Supervisory format responses**
RR—Receive Ready	RR—Receive Ready
RNR—Receive Not Ready	RNR—Receive Not Ready
REJ—Reject	REJ—Reject
Unnumbered format commands	**Unnumbered format responses**
UI—Unnumbered Information	UA—Unnumbered Acknowledgment
DISC—Disconnect	DM—Disconnected Mode
SABME—Set Asynchronous Balanced Mode Extended	FRMR—Frame Reject
XID—Exchange Identification	XID —Exchange Identification
TEST—Test	TEST—Test

5.4.1 Type 1 Operation Commands and Responses. The Type 1 commands and responses are all U-format PDU's.

5.4.1.1 Type 1 Operation Commands. The U-format PDU command encodings for Type 1 operation are listed in Fig 5-2:

FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER

1	2	3	4	5	6	7	8	
1	1	0	0	P	0	0	0	UI COMMAND
1	1	1	1	P	1	0	1	XID COMMAND
1	1	0	0	P	1	1	1	TEST COMMAND

Fig 5-2
Type 1 Operation Command Control Field Bit Assignments

5.4.1.1.1 Unnumbered Information (UI) Command. The UI command PDU shall be used to send information to one or more LLC's. Use of the UI command PDU is not dependent on the existence of a data link connection between the destination and source LLC's, and its use will not affect the V(S) or V(R) variables associated with any data link connections. There is no LLC response PDU to the UI command PDU.

Reception of the UI command PDU is not acknowledged or sequence number verified by the data link connection procedures; therefore, the UI PDU may be lost if a data link connection exception (such as a transmission error or a receiver-busy condition) occurs during the sending of the command PDU. A UI command PDU shall have either an individual, group, global, or null address as the destination DSAP address and the originator's individual address as the SSAP address.

5.4.1.1.2 Exchange Identification (XID) Command. The XID command PDU shall be used to convey the types of LLC services supported (for all LLC services) and the receive window size on a per data link connection basis to the destination LLC, and to cause the destination LLC to respond with the XID response PDU (see 5.4.1.2.1) at the earliest opportunity. The XID command PDU shall have no effect on any mode or sequence numbers maintained by the remote LLC. An XID command PDU shall have either an individual, group, global, or null address as the destination DSAP address and the originator's individual address as the SSAP address.

The information field of an XID basic format command PDU shall consist of an 8-bit XID format identifier field plus an 16-bit parameter field that is encoded to identify the LLC services supported plus the receive window size, as shown in Fig 5-3. The receive window size (k) is the maximum number that the send state variable V(S) can exceed the N(R) of the last received PDU.

NOTE: Other uses of the XID PDU are for further study. In particular, the use of an unsolicited XID response PDU to announce the presence of a new LLC will be examined.

5.4.1.1.3 Test (TEST) Command. The TEST command PDU shall be used to cause the destination LLC to respond with the TEST response PDU (see 5.4.1.2.2) at the earliest opportunity, thus performing a basic test of the LLC to LLC transmission path. An information field is optional with the TEST command

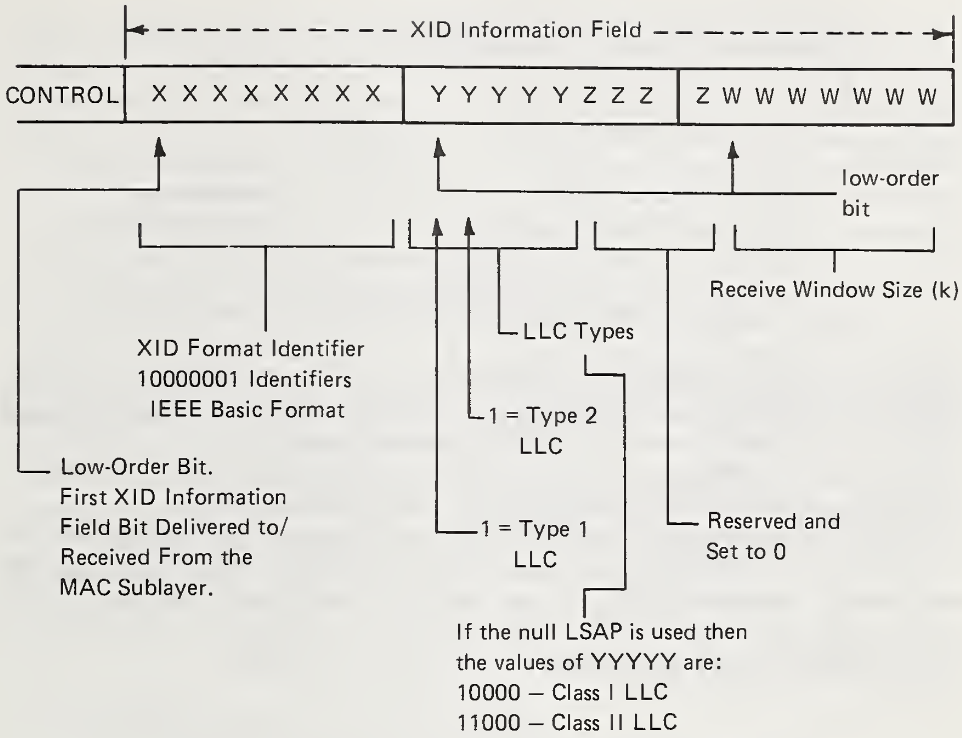


Fig 5-3
XID Information Field Basic Format

PDU. If present, however, the received information field shall be returned, if possible, by the addressed LLC in the TEST response PDU. The TEST command PDU shall have no effect on any mode or sequence numbers maintained by the remote LLC and may be used with an individual, group, global, or null DSAP address, and with an individual, group, or global DA address.

5.4.1.2 Type 1 Operation Responses. The U-format PDU response encodings for Type 1 operation are listed in Fig 5-4:

5.4.1.2.1 Exchange Identification (XID) Response. The XID response PDU shall be used to reply to an XID command PDU at the earliest opportunity. The XID response PDU shall identify the responding LLC and shall include an information field like that defined for the XID command PDU (see 5.4.1.1.2), regardless of what information is present in the information field of the received XID command PDU. The XID response PDU shall use an individual or null DSAP address and shall use an individual or null SSAP address. The XID response PDU shall have its F bit set to the state of the P bit in the XID command PDU.

5.4.1.2.2 Test (TEST) Response. The TEST response PDU shall be used to reply to the TEST command PDU. The TEST response PDU shall have its F bit set to the value of the P bit in the TEST command PDU. An information field, if present in the TEST command PDU, shall be returned in the corresponding

FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER

1	2	3	4	5	6	7	8	
1	1	1	1	F	1	0	1	XID RESPONSE
1	1	0	0	F	1	1	1	TEST RESPONSE

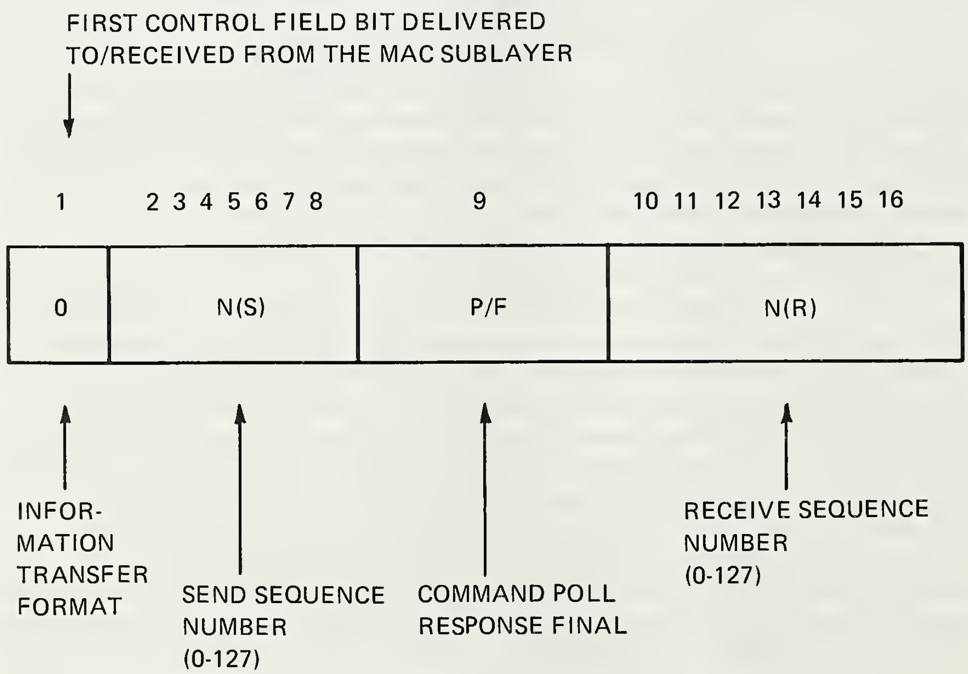
Fig 5-4
Type 1 Operation Response Control Field Bit Assignments

TEST response PDU. If the LLC cannot accept an information field, (eg, buffering limitation) a TEST response PDU without an information field may be returned. Refer to 6.7 for specific details on TEST response usage.

5.4.2 Type 2 Operation Commands and Responses. Type 2 commands and responses consist of I-format, S-format and U-format PDU's.

5.4.2.1 Information Transfer Format Command and Response. The function of the information, I, command and response shall be to transfer sequentially numbered PDU's containing an octet-oriented information field across a data link connection. The encoding of the I PDU control field for Type 2 operation shall be as listed in Fig 5-5.

Fig 5-5
Information Transfer Format Control Field Bits



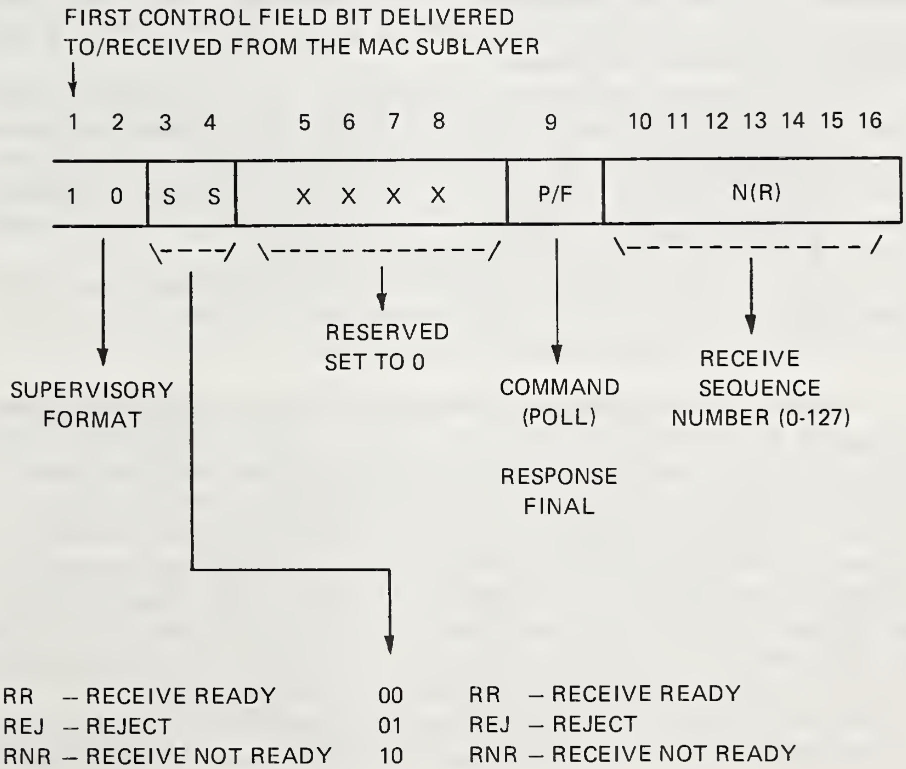
The I PDU control field shall contain two sequence numbers: N(S), send sequence number, which shall indicate the sequence number associated with the I PDU; and N(R), receive sequence number, which shall indicate the sequence number (as of the time the PDU is sent) of the next expected I PDU to be received, and consequently shall indicate that the I PDU's numbered up through N(R)-1 have been received correctly. See 5.3.2.3 for a description of P/F bit functions.

5.4.2.2 Supervisory Format Commands and Responses. Supervisory, S, PDU's shall be used to perform numbered supervisory functions such as acknowledgments, temporary suspension of information transfer, or error recovery.

PDU's with the S format shall not contain an information field and, therefore, shall not increment the send state variable at the sender or the receive state variable at the receiver. The encoding of the S-format PDU control field for Type 2 operation shall be as listed in Fig 5-6.

An S-format PDU shall contain an N(R), receive sequence number, which shall indicate, at the time of sending, the sequence number of the next expected I PDU to be received, and consequently shall indicate that all received I PDU's numbered up through N(R)-1 have been received correctly.

Fig 5-6
Supervisory Format Control Field Bits



When sent, an RR or REJ PDU shall indicate the clearance of any busy condition at the sending LLC that was indicated by the earlier sending of an RNR PDU. See 5.3.2.3 for a description of the P/F bit functions.

5.4.2.2.1 Receive Ready, (RR), Command and Response. The Receive Ready, RR, PDU shall be used by a LLC to indicate it is ready to receive an I PDU(s). I PDU's numbered up through N(R)-1 shall be considered as acknowledged.

5.4.2.2.2 Reject, (REJ), Command and Response. The Reject, REJ, PDU shall be used by a LLC to request the resending of I PDU's starting with the PDU numbered N(R). I PDU's numbered up through N(R)-1 shall be considered as acknowledged. It shall be possible to send additional I PDU's awaiting initial sending after the resent I PDU(s).

With respect to each direction of sending on a data link connection, only one "sent REJ" condition shall be established at any given time. The "sent REJ" condition shall be cleared upon the receipt of an I PDU with an N(S) equal to the N(R) of the REJ PDU. The "sent REJ" condition may be reset in accordance with procedures described in 7.5.4.

5.4.2.2.3 Receive Not Ready, (RNR), Command and Response. The Receive Not Ready, RNR, PDU shall be used by a LLC to indicate a busy condition (ie, a temporary inability to accept subsequent I PDU's). I, PDU's numbered up through N(R)-1 shall be considered as acknowledged. I PDU's numbered N(R) and any subsequent I PDU's received, if any, shall not be considered as acknowledged; the acceptance status of these PDU's shall be indicated in subsequent exchanges.

5.4.2.3 Unnumbered Format Commands and Responses. Unnumbered, U, commands and responses shall be used in Type 2, operation to extend the number of data link connection control functions. PDU's sent with the U format shall not increment the state variables on the data link connection at either the sending or the receiving LLC. The encoding of the U-format command/response PDU control field shall be as shown in Fig 5-7a.

The U-format command and response encodings for Type 2 operation are listed in Fig 5-7b.
See 5.3.2.3 for a description of the P/F bit functions.

5.4.2.3.1 Set Asynchronous Balanced Mode Extended (SABME) Command. The SABME command PDU shall be used to establish a data link connection to the destination LLC in the asynchronous balanced mode. No information shall be permitted with the SABME command PDU. The destination LLC shall confirm acceptance of the SABME command PDU by sending a UA response PDU on that data link connection at the earliest opportunity. Upon acceptance of the SABME command PDU, the destination LLC's send and receive state variables shall be set to zero. If the UA response PDU is received correctly, then the initiating LLC shall also assume the asynchronous balanced mode with its corresponding send and receive state variables set to zero.

Previously sent I PDU's that are unacknowledged when this command is actioned shall remain unacknowledged. Whether or not a LLC resends the

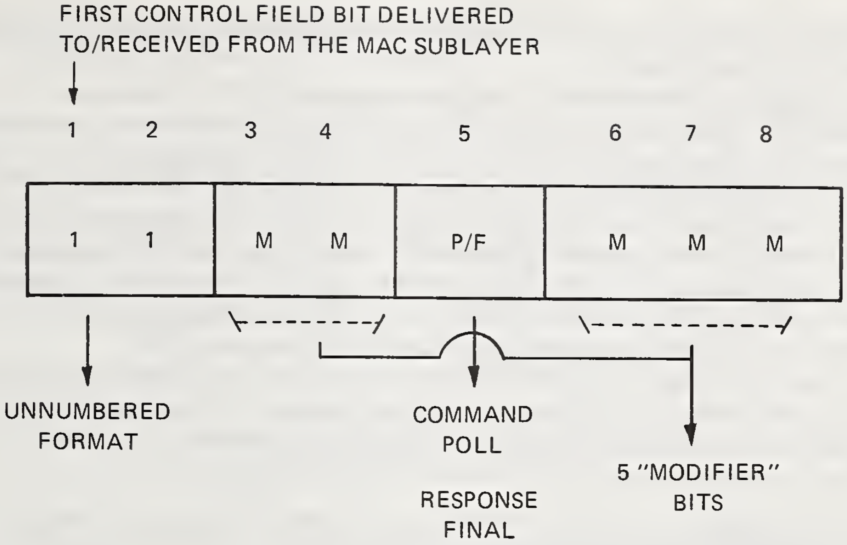


Fig 5-7a
Unnumbered Format Control Field Bits

FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER

1	2	3	4	5	6	7	8	
1	1	1	1	P	1	1	0	SABME COMMAND
1	1	0	0	P	0	1	0	DISC COMMAND
1	1	0	0	F	1	1	0	UA RESPONSE
1	1	1	1	F	0	0	0	DM RESPONSE
1	1	1	0	F	0	0	1	FRMR RESPONSE

Fig 5-7b
Unnumbered Command and Response Control Field Bit Assignments

contents of the information field of unacknowledged outstanding I PDU's shall be decided at a higher layer.

5.4.2.3.2 Disconnect (DISC) Command. The DISC command PDU shall be used to terminate an asynchronous balanced mode previously set by a SABME command PDU. It shall be used to inform the destination LLC that the source LLC is suspending operation of the data link connection and the destination LLC should assume the logically disconnected mode. No information field shall be permitted with the DISC command PDU. Prior to actioning the command the destination LLC shall confirm the acceptance of the DISC command PDU by sending a UA response PDU on that data link connection.

Previously sent I PDU's that are unacknowledged when this command is actioned shall remain unacknowledged. Whether or not a LLC resends the

contents of the information field of unacknowledged outstanding I PDU's shall be decided at a higher layer.

5.4.2.3.3 Unnumbered Acknowledgment (UA) Response. The UA response PDU shall be used by an LLC on a data link connection to acknowledge the receipt and acceptance of the SABME and DISC command PDU's. These received command PDU's shall not be actioned until the UA response PDU is sent. No information field shall be permitted with the UA response PDU.

5.4.2.3.4 Disconnect Mode (DM) Response. The DM response PDU shall be used to report status indicating that the LLC is logically disconnected from the data link connection and is, by system definition, in ADM. No information field shall be permitted with the DM response PDU.

5.4.2.3.5 Frame Reject (FRMR) Response. The FRMR response PDU shall be used by the LLC in the asynchronous balanced mode to report that one of the following conditions that is not correctable by resending the identical PDU resulted from the receipt of a PDU from the remote LLC:

(1) The receipt of a command PDU or a response PDU that is invalid or not implemented; examples of invalid PDU's include:

(a) the receipt of a supervisory or unnumbered PDU with an information field which is not permitted
(b) the receipt of an unsolicited F bit set to "1"
(c) the receipt of an unexpected UA response PDU

(2) The receipt of an I PDU, with an information field which exceeded the established maximum information field length that can be accommodated by the receiving LLC for that data link connection

(3) The receipt of an invalid N(R) from the remote LLC. [An invalid N(R) shall be defined as one which signifies an I PDU which has previously been sent and acknowledged, or signifies an I PDU which has not been sent and is not the next sequential I PDU awaiting to be sent.]

(4) The receipt of an invalid N(S) from the remote LLC. [An invalid N(S) shall be defined as an N(S) which is greater than or equal to the last sent N(R)+k, where k is the maximum number of outstanding I PDUS. The parameter k is the window size indicated in the XID PDU.]

The responding LLC shall send the FRMR response PDU at the earliest opportunity.

The LLC receiving the FRMR response PDU shall be responsible itself for initiating the appropriate mode setting or resetting corrective action by initializing both directions of transmissions on the data link connection, using the SABME and DISC command PDU's, as applicable.

An information field shall be returned with the FRMR response PDU to provide the reason for the PDU rejection. The information field shall contain the fields shown in Fig 5-8:

The functions of these fields shall be:

(1) Rejected PDU control field shall be the control field of the received PDU

FIRST CONTROL FIELD BIT DELIVERED
TO/RECEIVED FROM THE MAC SUBLAYER

1 - - - - - - - - - - - - - -16	17	18 - - 24	25	26 - - 32	33 - - 36	37 - - 40
REJECTED PDU CONTROL FIELD	0	V(S)	C/R	V(R)	WXYZ	V000

Fig 5-8
FRMR Information Field Format

which caused the FRMR exception condition on the data link connection. When the rejected PDU is a U format PDU, the control field of the rejected PDU shall be positioned in bit positions 1-8, with 9-16 set to 0.

(2) V(S) shall be the current send state variable value for this data link connection at the rejecting LLC (bit 18 = low-order bit).

(3) C/R set to "1" shall indicate that the PDU which caused the FRMR was a response PDU, and C/R set to "0" shall indicate that the PDU which caused the FRMR was a command PDU.

(4) V(R) shall be the current receive state variable value for this data link connection at the rejecting LLC (bit 26 = low-order bit).

(5) w set to "1" shall indicate that the control field received and returned in bits 1 through 16 was invalid or not implemented. Examples of invalid PDU are defined as:

(a) the receipt of a supervisory or unnumbered PDU with an information field which is not permitted;

(b) the receipt of an unsolicited F bit set to "1"; and

(c) the receipt of an unexpected UA response PDU.

(6) x set to "1" shall indicate that the control field received and returned in bits 1 through 16 was considered invalid because the PDU contained an information field which is not permitted with this command or response. Bit w shall be set to "1" in conjunction with this bit.

(7) y set to "1" shall indicate that the information field received exceeded the established maximum information field length which can be accommodated by the rejecting LLC on that data link connection.

(8) z set to "1" shall indicate that the control field received and returned in bits 1 through 16 contained an invalid N(R).

(9) v set to "1" shall indicate that the control field received and returned in bits 1 through 16 contained an invalid N(S). Bit w shall be set to "1" in conjunction with this bit.

6. LLC Description of the Type 1 Procedures

6.1 Modes of Operation. In Type 1 operation, no modes of operation are defined. An LLC using Type 1 procedures shall support the entire procedure set whenever it is operational on the local area network.

6.2 Procedure for Addressing. The address fields shall be used to indicate the source (SSAP) and destination (DSAP) of the LLC PDU. The first bit in the source address field (SSAP) shall be used to identify whether a command or a response is contained in the PDU.

Individual, group, global, and null addressing shall be supported for destination DSAP addresses. The source address field (SSAP) shall contain either an individual or null source address (see 3.3.1.2).

6.3 Procedure for the Use of the P/F Bit. A UI command PDU shall only be sent with the P bit set to "0." If a UI command PDU is received with the P bit set to "1," the LLC sublayer shall optionally discard it or pass it to the higher layer with a flag identifying that the P bit was set to "1." Since a UI PDU shall not be sent as a response PDU, procedures regarding the use of the F bit do not apply.

An XID command PDU shall have the P bit set to either "0" or "1." Upon receipt of an XID command PDU, the receiving LLC shall return an XID response PDU which has the F bit set equal to the value of the P bit contained in the incoming command PDU.

A TEST command PDU shall have the P bit set to either "0" or "1." Upon receipt of a TEST command PDU, the receiving LLC shall return a TEST response PDU which has the F bit set equal to the value of the P bit contained in the incoming command PDU.

6.4 Procedures for Logical Data Link Set-Up and Disconnection. Type 1 operation does not require any prior data link connection establishment (set-up), and hence no data link disconnection. Once the service access point has been enabled within the LLC, presumably by layer management's request, information may be sent to, or received from, a remote LLC service access point which is also participating in Type 1 operation.

6.5 Procedures for Information Transfer.

6.5.1 Sending UI PDU's. Information transfer shall be accomplished by sending the UI command PDU with the P bit set to "0." Sending UI PDU's with the P bit set to "1" or as response PDU's is prohibited. It shall be possible to send the UI command PDU at any time.

6.5.2 Receiving UI PDU's. Reception of the UI command PDU shall not be acknowledged or sequence number verified by the logical data link procedures; therefore, the UI PDU may be lost if a logical data link exception occurs during the sending of the command PDU. It shall be possible to receive a UI command PDU at any time. However, local conditions at the receiver may result in the discarding of valid UI command PDU's by the receiving LLC. UI command PDU's

which are received with the P bit set to "1" shall optionally be discarded or passed to the higher layer with a flag identifying that the P bit was set to "1."

UI PDU's which are response PDU's are invalid transmissions and shall be discarded by the receiving LLC.

6.6 Uses of the XID Command PDU and Response PDU. While the response to an XID command PDU shall be mandatory, the origination of an XID command PDU shall be optional. It shall be possible for the XID capabilities to be used as a part of some network control functions. As such, an XID command PDU may be sent on direction from a higher layer function, an administration function having access to the Data Link Layer, or an automatic start-up function. However, it shall also be possible for a more capable implementation of LLC to incorporate the use of the XID function directly to make more efficient use of the protocol.

Some possible uses of the XID capabilities include:

(1) The XID command PDU with a null LSAP is a way to solicit a response from any station (ie, any DA). As such it represents a basic "Are You There?" test capability.

(2) The XID command PDU with a group DA or group DSAP address can be used to determine the group membership. In particular, the XID command PDU with a global DA address can identify all active stations.

(3) A duplicate address check can be made (see Table 6-1a).

(4) For Class II LLC's in ABM, an XID exchange can be used to identify the receive window size at each LLC for that data link connection.

(5) An XID exchange with the null LSAP can identify each LLC's class.

(6) An XID exchange with separate LSAP's can identify the service types supported by those LSAP's.

(7) A LLC can announce its presence with a global DA address in an XID PDU.

6.7. Uses of the TEST Command PDU and Response PDU. The TEST function provides a facility to conduct loopback tests of the LLC to LLC transmission path. The initiation of the TEST function may be caused by an administration or management entity within the Data Link Layer. Successful completion of the test consists of sending a TEST command PDU with a particular information field provided by this administration or management entity to the designated destination LLC address and receiving in return the exact same information field in a TEST response PDU.

Implementation of the TEST command PDU is optional but every LLC must be able to respond to a received TEST command PDU with a TEST response PDU. The length of the information field is variable from 0 to the largest size specified that each LLC on this local area network must support for normal data transfer.

It shall also be possible to send even larger information fields with the following interpretations. If the receiving LLC can successfully receive and return the larger information field, it will do so. If it cannot receive the entire information field but the MAC can detect a satisfactory FCS, the LLC shall discard the portion of the information field received, and may return a TEST

Table 6-1a
Station Component State Transitions

Current State	Event	Action(s)	Next State
DOWN_STATE	[ENABLE_WITH_DUPLICATE_ADDRESS_CHECK]	SEND_NULL_DSAP_XID_C START_ACK_TIMER RETRY_COUNT:=0 XID_R_COUNT:=0	DUPLICATE_ADDRESS_CHECK_STATE
	[ENABLE_WITHOUT_DUPLICATE_ADDRESS_CHECK	REPORT_STATUS(STATION_UP)	UP_STATE
UP_STATE	DISABLE_REQUEST	REPORT_STATUS(STATION_DOWN)	DOWN_STATE
	RECEIVE_NULL_DSAP_XID_C	SEND_XID_R	UP_STATE
	RECEIVE_NULL_DSAP_TEST_C	SEND_TEST_R	UP_STATE
DUPLICATE_ADDRESS_CHECK_STATE	[RECEIVE_NULL_DSAP_XID R_AND_XID_R_COUNT=0]	XID_R_COUNT:=XID_R_COUNT+1	DUPLICATE_ADDRESS_CHECK_STATE
(OPTIONAL)	[RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=1	REPORT_STATUS(DUPLICATE_ADDRESS_FOUND)	DOWN_STATE
	[RECEIVE_NULL_DSAP_XID_C]	SEND_XID_R	DUPLICATE_ADDRESS CHECK_STATE
	[ACK_TIMER_EXPIRED_AND_RETRY_COUNT<MAXIMUM_RETRY]	SEND_SNULL_DSAP_XID_C START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1 XID_R_COUNT:=0	DUPLICATE_ADDRESS_CHECK_STATE
	[ACK_TIMER_EXPIRED_AND_RETRY_COUNT=MAXIMUM_RETRY	REPORT_STATUS(STATION_UP)	UP_STATE
	[DISABLE_REQUEST]	REPORT_STATUS(STATION_DOWN)	DOWN_STATE

response PDU with no information field. If the MAC cannot properly compute the FCS for the overlength information fields, the LLC shall discard the portion of the information field received, and shall give no response. Any TEST command PDU received in error shall be discarded and no response PDU sent.

In the event of failure, it shall be the responsibility of the administration or management entity which initiated the TEST function to determine any future actions.

6.8. List of Logical Data Link Parameters. A number of logical data link parameters are defined, the range of values for which are determined on a system-by-system basis by the user at the time that the local area network is established.

The logical data link parameters for Type 1 operation shall be as follows:

6.8.1. Maximum Number of Octets in a UI PDU. Refer to the appropriate MAC protocol specification for any limitation on the maximum number of octets in a UI PDU. No restrictions are imposed by the LLC sublayer. However, in the interest of having a value that all users of Type 1 LLC may depend upon, all

MAC's must at least be capable of accommodating UI PDU's with information fields up to and including 128 octets in length.

6.8.2. Minimum Number of Octets in a PDU. The minimum length valid PDU shall contain exactly two service access point address fields and one control field in that order. Thus the minimum number of octets in a valid PDU shall be 3.

6.9. Formal Description of the Type 1 Procedures. If discrepancies appear to exist with the text found in the balance of Section 6, this subsection (6.9) shall be viewed as being the definitive description.

6.9.1. LLC Formal Specification. The operation of the LLC is logically divided into several components. Each component characterizes a set of protocol operations performed by an LLC entity and is defined using a protocol state machine description. These state machines do not specify particular implementation techniques; rather, they are intended to describe the "external" characteristics of an LLC entity, as perceived by an LLC entity in a remote station or by a higher layer protocol in the local station.

The LLC operation is described using the following three types of components:

(1) **Station Component.** This component is responsible for processing the events which affect the entire LLC entity. The Station Component handles PDU's addressed to the null DSAP address and processes the duplicate address check, if implemented. One Station Component shall exist for each MAC Service Access Point present on the local area network.

(2) **Link Service Access Point (SAP) Component.** This component is responsible for processing the events which affect a specific operating service access point. One SAP Component shall exist for each SAP within the LLC entity.

(3) **Connection Component.** This component is responsible for processing the events which affect a specific data link connection for Type 2 procedures only (see 7.9 below). One Connection Component shall exist for each data link connection supported in the LLC entity.

The operation of each component is described using a state machine description. These important points are assumed in these descriptions:

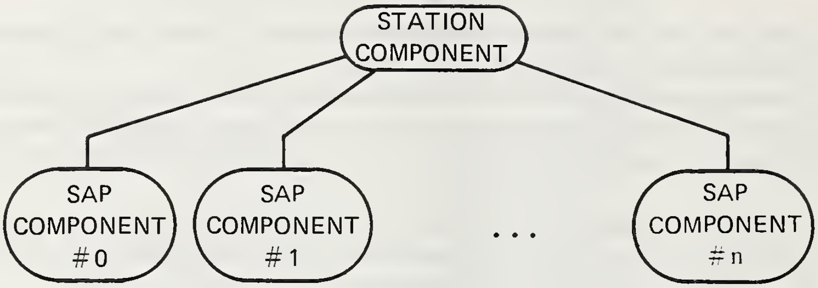
(a) The components are hierarchically related; ie, the Station Component is the "parent" of the SAP Component, which in turn is the "parent" of the Connection Component. See Fig 6-1.

(b) Each "parent" component has a state which provides the enabling conditions for its "child" component(s) to operate. If the parent component leaves this state, then the "child" component(s) are disabled.

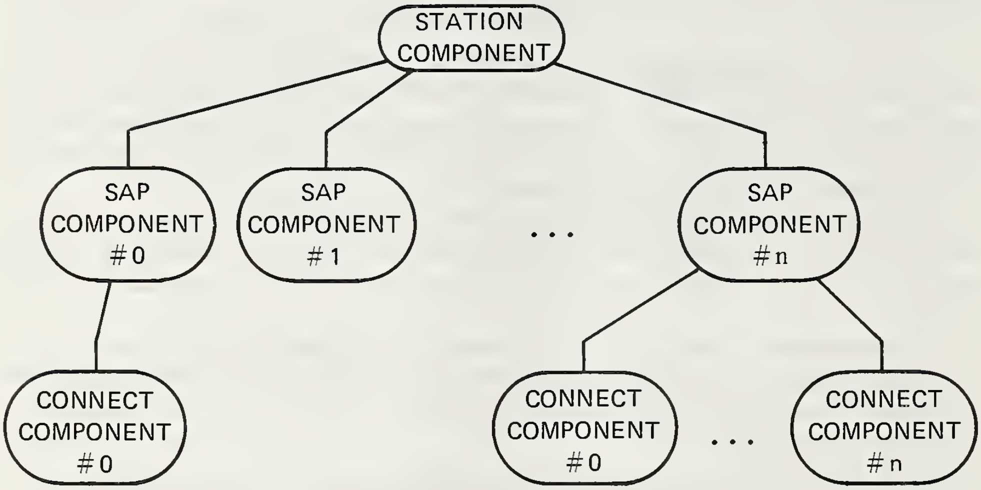
(c) For each "parent" component, several "child" components are allowed to be concurrently operating once their "parent" enabling conditions have been satisfied.

(d) There exists for each MAC Service Access Point one and only one LLC entity, consisting of the various operating components.

(e) In Class I LLC operation, each LLC can have zero or more SAP's being



(a) Class I LLC Component Relationship



(b) Class II LLC Component Relationship

Fig 6-1
Component Relationships

serviced (ie, active) at any one time, independent of each other, which are differentiated by the DSAP address. The services of a SAP shall be provided by a separate Service Access Point Component.

(f) In Class II LLC operation, the services of each SAP can also support zero or more concurrent data link connections (each designated by the logical concatenation of the MAC address (DA/SA) and the LLC address (DSAP/SSAP). Each data link connection is controlled by a separate Connection Component.

Each Component description shall consist of these sections:

(i) **Component Overview.** This section shall discuss the overall purpose behind the component operation.

(ii) **Component State Transition Diagram.** This diagram shall graphically represent the component machine overview.

(iii) **Component State Transition Table.** This chart shall display a table of the state transitions, including columns for current state, event, action(s), and next state. This table shall define all valid events for each state as well as the resultant component action(s) and state change.

(iv) **Component Event Description.** Each of the events which are used in the state transition table is explained.

(v) **Component Action Description.** Each of the actions which are used in the state transition table is explained.

(vi) **Component State Description.** Each of the states that are used in the state transition table are explained.

The following basic state machine operation rules apply:

(A) Events shall cause a state transition in the machine, and shall result in execution of some action(s) along with a state change (which may return to the same state).

(B) Events which are not listed as valid inputs to the current state of any of the operating components shall not cause:

(I) actions or state changes; or

(II) PDU transmissions.

The station should perform some error recovery which is appropriate for the particular implementation.

(C) If an incoming PDU is destined for a DSAP which is not active (ie, the appropriate component is not operating), it shall be considered to be an exception and dealt with in a manner appropriate for the receiving station.

6.9.2 Station Component Overview. The Station Component is responsible for handling all events that are directed to the LLC as a whole (ie, events affecting all SAP's and connections serviced by that LLC). The Station Component shall begin in the DOWN state, optionally check for a duplicate station address, and potentially enter the UP state; see Fig 6-2 and Table 6-1a. The UP state of the LLC Station Component provides the enabling conditions for the operation of the Service Access Point (SAP) Components.

The Station Component shall be capable of receiving and responding to the XID and TEST command PDU's. It shall optionally be capable of initiating the XID command PDU, if duplicate address checking is performed by the LLC entity in a particular implementation; see Table 6-1b. These PDU's shall use the null DSAP address to denote the Station Component is being referenced.

The performance of the duplicate address check requires the Station Component be prepared to receive its own XID PDU's. The definition of the MAC operation provides for the ability to simultaneously transmit and receive. Since the DA=SA in the XID PDU's can be used for duplicate address check, the MAC

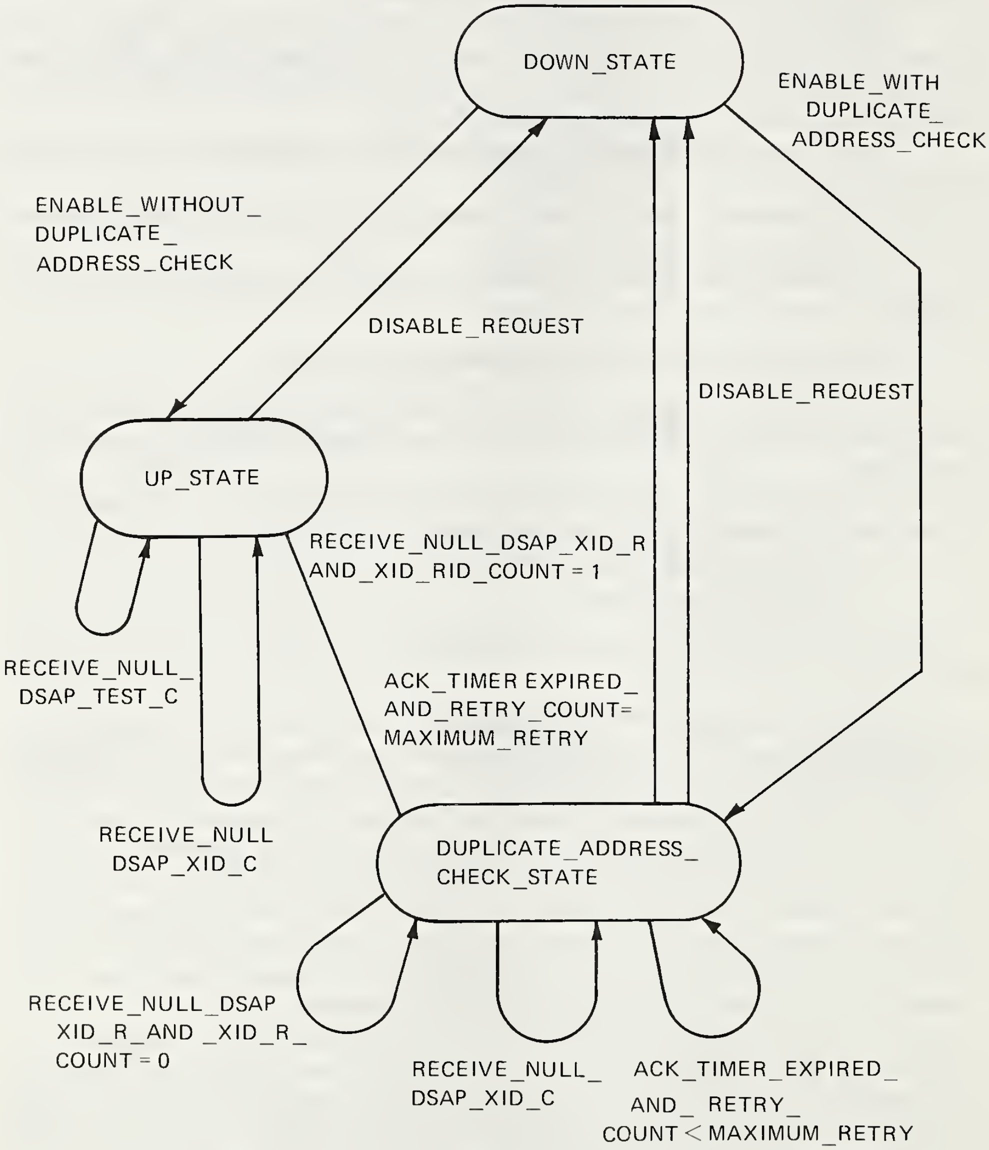


Fig 6-2
Station Component State Diagram

Table 6-1b
Station Component Options

Description	States Omitted	Other Requirements
No duplicate address check	DUPLICATE_ADDRESS_ CHECK_STATE	Omit: ENABLE_WITH_DUPLICATE_ ADDRESS_CHECK ACK_TIMER_EXPIRED_AND_ RETRY_COUNT<MAXIMUM_ RETRY ACK_TIMER_EXPIRED_AND_ RETRY_COUNT=MAXIMUM_ RETRY RECEIVE_NULL_DSAP_ XID_R_AND_XID_R_ COUNT=1 RECEIVE_NULL_DSAP_ XID_R_AND_XID_R_ COUNT=0
Optional use of duplicate address check	none	Omit: none
Always perform duplicate address check	none	Omit: ENABLE_WITHOUT_ DUPLICATE_ADDRESS_ CHECK

will recognize its own address and pass the PDU to the Station Component. The Station Component will respond to an XID command PDU with an XID response PDU, regardless of whether it originated from itself or a remote LLC. The Station Component provides the duplicate address check by maintaining a count of received XID response PDU's. If more than one XID response PDU is received, then at least one other identical MAC DA exists on the LAN. See Fig 6-2 and Table 6-1a for details.

6.9.2.1 Station Component State Description.

(1) **DOWN_STATE.** The Station Component is powered off, not initialized, and/or disabled from operating in the local area network.

(2) **DUPLICATE_ADDRESS_CHECK_STATE.** The Station Component is in process of checking for duplicate MAC addresses on the LAN. The main purpose of this state shall be to allow the LLC Station Component to verify that this station's MAC address is unique on the LAN. The Station Component shall send XID command PDU's with identical MAC DA and SA addresses, and shall wait for a possible XID Response PDU indicating the existence of other stations with identical MAC link addresses.

(3) **UP_STATE.** The Station Component is enabled, powered on, initialized, and operating in the local area network. The LLC shall allow SAP's to exchange LLC PDU's on the medium.

6.9.2.2 Station Component Event Description.

(1) **ENABLE_WITH_DUPLICATE_ADDRESS_CHECK.** Station Component user has initialized/enabled the station equipment, and has requested that the LLC check for MAC service access point address duplications before participating in data link communications.

(2) **ENABLE_WITHOUT_DUPLICATE_ADDRESS_CHECK.** Station Component user has initialized/enabled the equipment, but duplicate MAC service access point address checking by the LLC is not supported/desired.

(3) **ACK_TIMER_EXPIRED_AND_RETRY_COUNT<MAXIMUM _RETRY.** Acknowledgment timer has expired and retry count is less than maximum retry limit.

(4) **ACK_TIMER_EXPIRED_AND_RETRY_COUNT=MAXIMUM _RETRY.** Acknowledgment timer has expired and retry count is equal to the maximum retry limit.

(5) **RECEIVE_NULL_DSAP_XID_C.** An XID Command PDU with the null DSAP address has been received.

(6) **RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=0.** A single XID response PDU with the null DSAP address has been received.

(7) **RECEIVE_NULL_DSAP_XID_R_AND_XID_R_COUNT=1.** A second XID response PDU with the null DSAP address have been received.

(8) **RECEIVE_NULL_DSAP_TEST_C.** A TEST command PDU with the null DSAP address has been received.

(9) **DISABLE_REQUEST.** Station user has requested that the equipment be disabled from operating on the medium.

6.9.2.3 Station Component Action Description.

(1) **START_ACK_TIMER.** Start the acknowledgment timer. This allows the LLC to determine that it has not received an acknowledgment from the remote station within a specified response time.

(2) **RETRY_COUNT:=0.** Initialize the retry counter.

(3) **RETRY_COUNT:=RETRY_COUNT+1.** Increment the retry counter.

(4) **XID_R_COUNT:=0.** Initialize the XID response PDU counter.

(5) **XID_R_COUNT:=XID_R_COUNT+1.** Increment the XID response PDU counter.

(6) **SEND_NULL_DSAP_XID_C.** The LLC shall send an XID command PDU with null SSAP and null DSAP addresses and with identical MAC DA and SA addresses.

(7) **SEND_XID_R.** The LLC shall send an XID response PDU, using the SSAP address of the XID command PDU as the DSAP address of the response PDU, and using a null SSAP address.

(8) **SEND_TEST_R.** The LLC shall send a TEST response PDU, using the SSAP address of the TEST command PDU as the DSAP address of the response PDU, and using a null SSAP address.

(9) **REPORT_STATUS.** The LLC shall be able to report data link status conditions, with the following valid reasons:

(a) **STATION_UP.** LLC entity is now operational.

(b) **STATION_DOWN.** The LLC entity is now non-operational.

(c) **DUPLICATE_ADDRESS_FOUND.** LLC entity has detected another LLC entity on the LAN with a MAC service access point address identical to its own.

6.9.3 Service Access Point (SAP) Component Overview. The Service Access Point (SAP) Component handles all LLC Type 1 PDU traffic for a particular DSAP address in the local Station Component. The local service access point user is able to activate and deactivate the operation of each individual SAP Component in the Station Component (see Fig 6-3 and Table 6-2). Once active, the SAP Component shall process Type 1 LLC PDU's addressed to the DSAP and send Type 1 LLC PDU's either by service access point user request or as a result of some LLC protocol action.

For Class II stations, the ACTIVE state of the SAP Component provides the activating conditions for Type 2 LLC Connection Component services (see Fig 6-1). Any attempt to make a data link connection, either by the user or a remote LLC, while the SAP component is ACTIVE shall be passed to the Type 2 LLC Connection Component and ignored by the SAP Component (this includes the handling of the Disconnect Mode for a Type 2 LLC Connection Component).

Table 6-2
Service Access Point Component State Transitions

Current State	Event	Action(s)	Next State
INACTIVE_ STATE	SAP_ACTIVATION_REQUEST	REPORT_STATUS(SAP_ACTIVE)	ACTIVE_STATE
ACTIVE_STATE	RECEIVE_UI	DATA_INDICATE	ACTIVE_STATE
	DATA_REQUEST	SEND_UI	ACTIVE_STATE
	XID_REQUEST	SEND_XID_C	ACTIVE_STATE
	RECEIVE_XID_C	SEND_XID_R	ACTIVE_STATE
	RECEIVE_XID_R	XID_INDICATE	ACTIVE_STATE
	TEST_REQUEST	SEND_TEST_C	ACTIVE_STATE
	RECEIVE_TEST_C	SEND_TEST_R	ACTIVE_STATE
	RECEIVE_TEST_R	TEST_INDICATE	ACTIVE_STATE
	SAP_DEACTIVATION_REQUEST	REPORT_STATUS(SAP_INACTIVE)	INACTIVE_ STATE

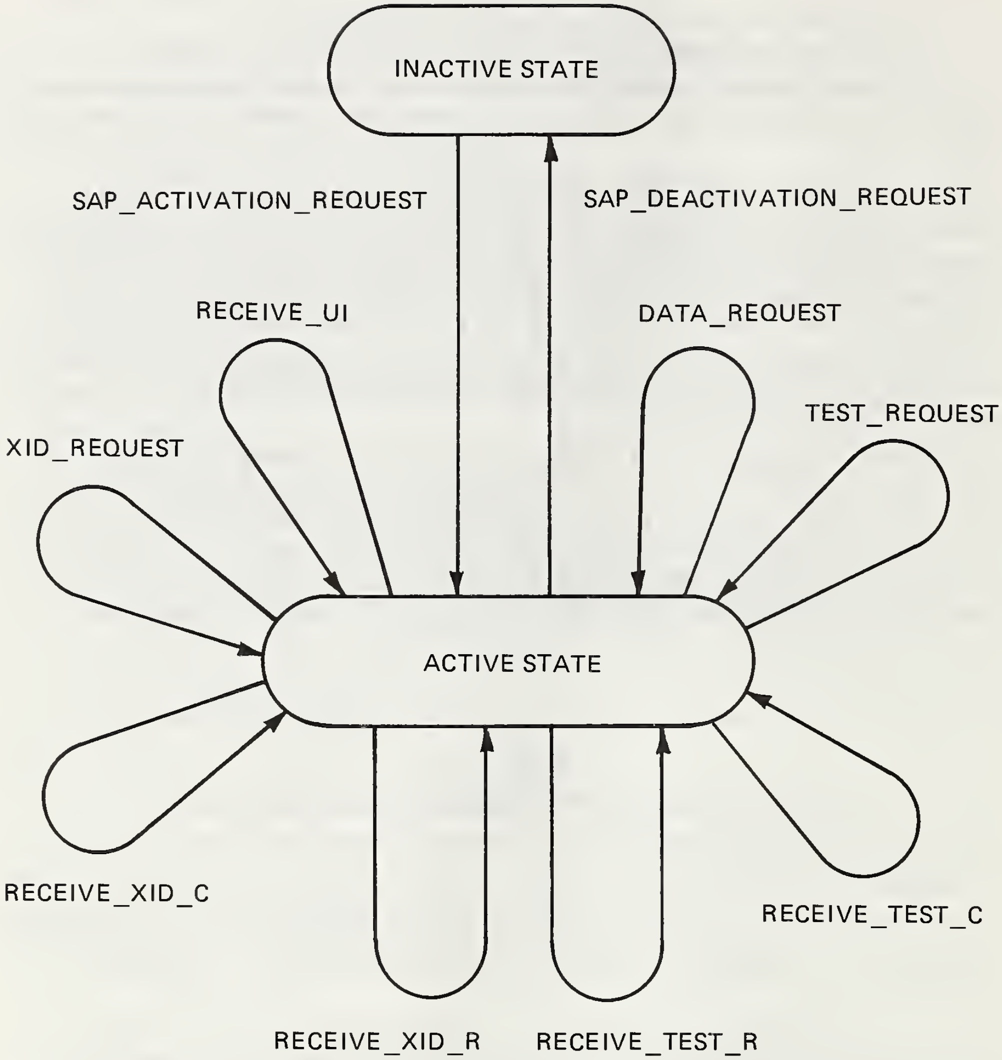


Fig 6-3
Link Service Access Point State Diagram

6.9.3.1 Service Access Point (SAP) Component State Descriptions.

(1) **INACTIVE_STATE.** LLC SAP Component is not active, functioning, or operational. No PDU's are accepted and/or sent.

(2) **ACTIVE_STATE.** LLC SAP Component is active, functioning, and operational. PDU's are received and sent.

6.9.3.2 Service Access Point (SAP) Component Event Description.

(1) **SAP_ACTIVATION_REQUEST.** The SAP user has requested that

the particular LLC SAP Component be activated and begin logical data link operation of the Type 1 services.

(2) **SAP_DEACTIVATION_REQUEST.** The SAP user has requested that the particular LLC SAP Component be deactivated and no longer allowed to operate on the logical data link.

(3) **XID_REQUEST.** The SAP user has requested that the LLC SAP Component send an XID command PDU to one or more remote SAP's.

(4) **TEST_REQUEST.** The SAP user has requested that the LLC SAP Component send a TEST command PDU to one or more remote SAP's.

(5) **RECEIVE_UI.** The local SAP Component has received a UI PDU from a remote SAP.

(6) **DATA_REQUEST.** The SAP user has requested that a Data Unit be passed to a remote LLC SAP, via an UI PDU.

(7) **RECEIVE_XID_C.** The local SAP Component has received an XID command PDU from a remote SAP.

(8) **RECEIVE_XID_R.** The local SAP Component has received an XID response PDU from a remote SAP.

(9) **RECEIVE_TEST_C.** The Local SAP Component has received a TEST command PDU from the remote SAP.

(10) **RECEIVE_TEST_R.** The local SAP Component has received a TEST response PDU from the remote SAP.

6.9.3.3 Service Access Point (SAP) Component Action Description.

(1) **DATA_INDICATE.** LLC SAP Component has received a UI PDU from a remote SAP. The service data unit is given to the SAP user.

(2) **SEND_UI.** A UI PDU is send to one or more remote SAP's in response to a user request to send a service data unit.

(3) **SEND_XID_C.** LLC SAP Component shall send an XID command PDU to remote SAPs in response to a SAP user request to identify other SAP's.

(4) **SEND_XID_R.** LLC SAP Component shall send an XID response PDU to remote SAP's in response to a received XID command PDU.

(5) **SEND_TEST_C.** LLC SAP Component shall send a TEST command PDU in response to SAP user request to test remote SAP.

(6) **SEND_TEST_R.** LLC SAP Component shall send a TEST response PDU in response to a remote LLC TEST command PDU.

(7) **REPORT_STATUS.** The LLC SAP Component shall be able to report data link status conditions for the particular SAP Component with the following valid reasons:

(a) **SAP_ACTIVE.** The SAP_ACTIVATION_REQUEST has been successfully processed and the component is reporting that it is now operational.

(b) **SAP_INACTIVE.** The SAP_DEACTIVATION_REQUEST has been successfully processed and the component is now deactivated.

(8) **XID_INDICATE.** LLC SAP Component has received an XID response PDU from a remote SAP. An indication of this event is passed to the SAP user, and may also return the XID information field.

(9) **TEST_INDICATE.** LLC SAP Component has received a TEST response PDU from a remote SAP. An indication of this event is passed to the SAP user, and may also return the TEST information field.

7. LLC Description of the Type 2 Procedures

7.1 Modes. In Type 2 operation, two modes of operation are defined, an operational mode and a non-operational mode.

7.1.1 Operational Mode. The one operational mode shall be the asynchronous balanced mode (ABM).

ABM is a balanced operational mode where a data link connection has been established between two service access points. Either LLC shall be able to send commands at any time and initiate response transmissions without receiving explicit permission from the other LLC. Such an asynchronous transmission shall contain one or more LLC PDU and shall be used for information field transfer and/or to indicate status changes in the LLC (for example, the number of the next expected information LLC PDU, transition from a ready to a busy condition or vice versa, occurrence of an exception condition).

ABM consists of a Data Link Connection Phase, an Information Transfer Phase, a Data Link Resetting Phase, and a Data Link Disconnection Phase.

7.1.2 Non-operational Mode. The one non-operational mode shall be the asynchronous disconnected mode (ADM).

ADM differs from the operational mode (ABM) in that the data link connection is logically disconnected from the physical medium; ie, no information (user data) shall be sent or accepted.

ADM is defined to prevent a data link connection from appearing on the physical medium in a fully operational mode during unusual situations or exception conditions since such operation could cause:

(1) sequence number mismatch between the LLC's on the data link connection, or

(2) ambiguity in one LLC as to another LLC's status.

A data link connection shall be system predefined as to the condition(s) that cause it to assume the asynchronous disconnected mode (ADM).

Examples of possible conditions (in addition to receiving a DISC command PDU) which shall cause a data link connection to enter ADM are:

(a) the power is turned on

(b) the Data Link Layer logic is manually reset

(c) the data link connection is manually switched from a local (home) condition to the connected-on-the-data-link (on-line) condition

A LLC on a data link connection in ADM shall be required to monitor transmissions received from its MAC for the purpose of:

(i) accepting and responding to one of the mode setting command PDU (SABME, DISC), or

(ii) sending a DM response PDU at a medium access opportunity, when required.

In addition, since the LLC has the ability to send command PDU's at any time, the LLC may send an appropriate mode setting command PDU.

An LLC in ADM receiving a DISC command PDU shall respond with the DM response PDU. An LLC in ABM receiving a DISC command PDU shall respond with the unnumbered acknowledgment (UA) response PDU if it is capable of actioning the command.

An LLC in ADM shall not establish a frame reject exception condition (see 5.4.2.3.5 and 7.6). ADM consists of a Data Link Disconnected Phase.

7.2 Procedure for Addressing. The address fields shall be used to indicate the source (SSAP) and destination (DSAP) of the PDU. The first bit in the source address field (SSAP) shall be used to identify whether a command or response is contained in the PDU.

A single data link connection can be established between any two service access points on the local area network. This data link connection is identified by a pair of "complete" data link addresses, each of which consists of a logical concatenation of the implicit physical address (not contained in the frame structure), the MAC address (DA/SA), and the LLC address (DSAP/SSAP). In order for a receiving DSAP to correctly identify the data link connection associated with an incoming PDU, the receiving DSAP must have access to the "complete" data link address information for the remote service access point.

7.3 Procedures for the Use of the P/F Bit. The LLC receiving a command PDU (SABME, DISC, RR, RNR, REJ, or I) with the P bit set to "1," shall send a response PDU with the F bit set to "1."

The response PDU returned by a LLC to a SABME or DISC command PDU with the P bit set to "1" shall be a UA or DM response PDU with the F bit set to "1." The response PDU returned by an LLC to an I, RR, or REJ command PDU with the P bit set to "1" shall be an I, RR, REJ, RNR, DM, or FRMR response PDU with the F bit set to "1." The response PDU returned by an LLC to an RNR command PDU with the P bit set to "1" shall be an RR, REJ, RNR, DM, or FRMR response PDU with the F bit set to "1."

NOTE: The P bit is usable by the LLC in conjunction with the timer recovery condition (see 7.5.9 below).

7.4 Procedures for Data Link Set-Up and Disconnection.

7.4.1 Data Link Connection Phase. Either LLC shall be able to take the initiative to initialize the data link connection.

When the LLC wishes to initialize the link, it shall send the SABME command PDU and start the Acknowledgment Timer (see 7.8.1 below). Upon reception of the UA response PDU, the LLC shall have reset both its send and receive state variables V (S) and V (R) to 0 for the corresponding data link connection, shall stop its Acknowledgment Timer, and shall enter the information transfer phase.

When receiving the DM response PDU, the LLC which originated the SABME command PDU shall stop its Acknowledgment Timer, shall not enter the information transfer phase, and shall report to the higher layer for appropriate action.

For a description of the actions to be followed upon receipt of a SABME or DISC command PDU, see 7.4.5. When receiving any other Type 2 command PDU with the P bit set to "1," the LLC which is attempting to establish a connection shall send a DM response PDU with the F bit set to "1." Other Type 2 PDU's received (commands and responses) while attempting to connect shall be ignored by the LLC.

Should the Acknowledgement Timer run out before reception of the UA or DM response PDU, the LLC shall resend the SABME command PDU and restart the Acknowledgment Timer. After resending the SABME command PDU N2 times, the sending LLC shall stop sending the SABME command PDU and shall report to the higher layer for the appropriate error recovery action to initiate. The value of N2 is defined in 7.8.2 below.

Whenever receiving an SABME command PDU, the LLC shall return a UA response PDU to the remote LLC and set both its send and receive state variables V(S) and V(R) to 0 for the corresponding data link connection and enter the information transfer phase. The return of the UA response PDU shall take precedence over any other response PDU for the same data link connection which may be pending at the LLC. It shall be possible to follow the UA response PDU with additional LLC PDU's, if pending.

If, upon receipt of the SABME command PDU, the LLC determines that it cannot enter the indicated phase, it shall send the DM response PDU and shall remain in the link disconnected mode.

7.4.2 Information Transfer Phase. After having sent the UA response PDU to an SABME command PDU, or having received the UA response PDU to a sent SABME command PDU, the LLC shall accept and send I-format and S-format PDU's according to the procedures described in 7.5 below.

When receiving an SABME command PDU while in the information transfer phase, the LLC shall conform to the resetting procedure described in 7.6.

7.4.3 Data Link Disconnection Phase. During the information transfer phase, either LLC shall be able to initiate disconnecting of the data link connection by sending a DISC command PDU.

When the LLC wishes to disconnect the data link connection, it shall send the DISC command PDU and start the Acknowledgment Timer (see 7.8.1). Upon reception of the UA or DM response PDU from the remote LLC, the LLC shall stop its Acknowledgment Timer and enter the link disconnected mode.

Should the Acknowledgment Timer run out before reception of the UA or DM response PDU, the LLC shall resend the DISC command PDU and restart the Acknowledgment Timer. After sending the DISC command PDU N2 times, the sending LLC shall stop sending the DISC command PDU, shall enter the data link disconnected phase, and shall report to the higher layer for the appropriate error recovery action to initiate. The value of N2 is defined in 7.8.2.

When receiving a DISC command PDU, the LLC shall return a UA response

PDU and enter the data link disconnected phase. The return of the UA response PDU shall take precedence over any other response PDU for the same data link connection which may be pending at the LLC.

7.4.4 Data Link Disconnected Phase. After having received a DISC command PDU from the remote LLC and returned a UA response PDU, or having received the UA response PDU to a sent DISC command PDU, the LLC shall enter the data link disconnected phase.

In the disconnected phase, the LLC shall be able to initiate data link connection. In the disconnected phase, the LLC shall react to the receipt of an SABME command PDU as described in 7.4.1 above and shall send a DM response PDU in answer to a received DISC command PDU.

When receiving any other Type 2 command PDU with the P bit set to "1" in the disconnected phase, the LLC shall send a DM response PDU with the F bit set to "1." Other Type 2 PDU's received in the disconnected phase shall be ignored by the LLC.

7.4.5 Contention of Unnumbered Mode Setting Command PDU's. A contention situation in a LLC shall be resolved in the following way.

If the sent and received mode-setting command PDU's are the same, each LLC shall send the UA response PDU at the earliest opportunity. Each LLC shall enter the indicated phase either after receiving the UA response PDU, or after its Acknowledgment Timer expires.

If the sent and received mode-setting command PDU's are different, each LLC shall enter the data link disconnected phase and shall issue a DM response PDU at the earliest opportunity.

7.5 Procedures for Information Transfer. The procedures which apply to the transfer of I PDU's in each direction on a data link connection during the information transfer phase are described below.

In the following, "number one higher" is in reference to a continuously repeated sequence series, ie, 127 is one higher than 126 and 0 is one higher than 127 for modulo 128 series.

7.5.1 Sending I PDU's. When the LLC has an I PDU to send (ie, an I PDU not already sent, or having to be resent as described in 7.5.5 below), it shall send the I PDU with an N(S) equal to its current send state variable V(S), and an N(R) equal to its current receive state variable V(R) for that data link connection. At the end of sending the I PDU, the LLC shall increment its send state variable V(S) by one.

If the Acknowledgment Timer is not running at the time that an I PDU is sent, the Acknowledgment Timer shall be started.

If the data link connection send state variable V(S) is equal to the last value of N(R) received plus k (where k is the maximum number of outstanding I PDU's, see 7.8.4) the LLC shall not send any new I PDU's on that data link connection, but shall be able to resend an I PDU as described in 7.5.6 or 7.5.9.

When a local LLC data link connection is in the busy condition, the LLC shall still be able to send I PDU's, provided that the remote LLC on this data link connection is not busy itself. When the LLC for a particular data link connection

is in the FRMR exception condition, it shall stop transmitting I PDU's on that data link connection.

7.5.2 Receiving an I PDU. When the LLC data link connection is not in a busy condition and receives an I PDU whose send sequence number is equal to the receive state variable V(R), the LLC shall accept the information field of this PDU, increment by one its receive state variable V(R), and act as follows:

(1) If an I PDU is available to be sent, the LLC shall be able to act as in 7.5.1 above and acknowledge the received I PDU by setting N(R) in the control field of the next sent I PDU to the value of the receive state variable V(R). The LLC shall also be able to acknowledge the received I PDU by sending an RR PDU with the N(R) equal to the value of the receive state variable V(R).

(2) If no I PDU is available to be sent by the LLC, then the LLC shall either:

(a) send an RR PDU with the N(R) equal to the value of the receive state variable V(R) at the earliest opportunity; or

(b) if the received PDU was *not* a command PDU with the P bit set to "1," wait for some period of time bounded by the probability of the remote Acknowledgment Timer expiry, for either an I PDU to become available for transmission, or to accumulate additional I PDU's to be acknowledged in a single RR PDU, subject to window size constraints.

(3) If receipt of the I PDU caused the LLC to go into the busy condition with regard to any subsequent I PDU's, the LLC shall send an RNR PDU with the N(R) equal to the value of the receive state variable V(R). If an I PDU(s) is available to send, the LLC shall be able to send them as in 7.5.1 above prior to or following the sending of the RNR PDU.

When the LLC associated with a particular data link connection is in a busy condition, and receives an in-sequence I PDU, the LLC shall be able to ignore the information field contained in any received I PDU on that data link connection (see 7.5.8).

7.5.3 Reception of Incorrect PDU's. When the LLC receives an invalid PDU (see 3.3.5) or a PDU with an incorrect DSAP or SSAP address, this PDU shall be discarded entirely.

7.5.4 Reception of Out-of-Sequence PDU's. When the LLC receives an I PDU whose send sequence number is not in sequence, ie, not equal to the current receive state variable V(R) but is within the receive window, the LLC shall discard the information field of the I PDU and send a REJ PDU with the N(R) set to the value of V(R). The LLC shall then discard the information field of all I PDU's until the expected I PDU is correctly received. When receiving the expected I PDU, the LLC shall acknowledge the PDU as described in 7.5.2 above. The LLC shall use the N(R) and P bit indications in the discarded I PDU's.

On a given data link connection, only one "sent REJ" exception condition from a given LLC to another given LLC shall be established at a time. A "sent REJ" condition shall be cleared when the requested I PDU is received. The "sent REJ" condition shall be able to be reset when a Reject Timer time-out function runs out. When the LLC perceives by Reject Timer time-out that the requested I PDU will not be received, because either the requested I PDU or the REJ PDU was in

error or lost, the LLC shall be able to repeat the REJ PDU in order to re-establish the "sent REJ" condition up to N2 times. The value of N2 is defined in 7.8.2.

7.5.5 Receiving Acknowledgment. When correctly receiving an I-format or S-format PDU, even in the busy condition (see 7.5.8), the receiving LLC shall consider the N(R) contained in this PDU as an acknowledgment for all the I PDU's it has sent on this data link connection with an N(S) up to and including the received N(R) minus one. The LLC shall reset the Acknowledgment Timer when it correctly receives an I-format or S-format PDU with the N(R) higher than the last received N(R) (actually acknowledging some I PDU's).

If the timer has been reset and there are outstanding I PDU's still unacknowledged on this data link connection, the LLC shall restart the Acknowledgment Timer. If the timer then runs out, the LLC shall follow the resending procedure (in 7.5.6 and 7.5.9) with respect to the unacknowledged I PDU's.

7.5.6 Receiving a REJ PDU. When receiving a REJ PDU, the LLC shall set its send state variable V(S) to the N(R) received in the REJ PDU control field. The LLC shall (re)send the corresponding I PDU as soon as it is available. If other unacknowledged I PDU's had already been sent on that data link connection following the one indicated in the REJ PDU, then those I PDU's shall be resent by the LLC following the resending of the requested I PDU.

7.5.7 Receiving an RNR PDU. A LLC receiving an RNR PDU shall stop sending I PDU's on the indicated data link connection at the earliest possible time, and shall start the Busy-State Timer, if not already running. When the Busy-State Timer runs out, the LLC shall follow the procedure described in 7.5.9. In any case the LLC shall not send any other I PDU's on that data link connection before receiving an RR or REJ PDU, or before receiving an I response PDU with the F bit set to "1," or before the completion of a resetting procedure on that data link connection.

7.5.8 LLC Busy Condition. A LLC shall enter the busy condition on a data link connection when it is temporarily unable to receive or continue to receive I PDU's due to internal constraints; for example, receive buffering limitations. When the LLC enters the busy condition, it shall send a RNR PDU at the earliest opportunity. It shall be possible to send I PDU's awaiting to be sent on that data link connection prior to or following the sending of the RNR PDU. While in the busy condition, the LLC shall accept and process supervisory PDU's and return an RNR response PDU with the F bit set to "1" if it receives a supervisory or I command PDU with the P bit set to "1" on the affected data link connection.

To indicate the clearance of a busy condition on a data link connection, the LLC shall send either a I response PDU with the F bit set to "1" if a P bit set to "1" is outstanding, a REJ response PDU, or a RR response PDU on the data link connection with N(R) set to the current receive state variable V(R), depending on whether or not the LLC discarded information fields of correctly received I PDU's. Additionally, the sending of a SABME command PDU or a UA response PDU shall indicate the clearance of a busy condition at the sending LLC on a data link connection.

7.5.9 Waiting Acknowledgment. The LLC maintains an internal retransmission count variable for each data link connection which shall be set to "0"

when the LLC receives or sends a UA response PDU to an SABME command PDU, or when the LLC receives a RNR frame PDU, or when the LLC correctly receives an I-format or S-format PDU with the N(R) higher than the last received N(R) (actually acknowledging some outstanding I PDU's).

If the Acknowledgment Timer runs out, the LLC on this data link connection shall enter the timer recovery condition, add one to its retransmission count variable and sét an internal variable X to the current value of its send state variable.

The LLC shall then start the P-bit Timer, set its send state variable to the last N(R) received from the remote LLC on this data link connection and send a S-format command PDU with the P bit set to "1."

The timer recovery condition shall be cleared on the data link connection when the LLC receives a valid I-format or S-format PDU from the remote LLC, with the F bit set to "1."

If, while in the timer recovery condition, the LLC correctly receives a valid I-format or S-format PDU with the F bit set to "1" and with the N(R) within the range from its current send state variable to X included, the LLC shall clear the timer recovery condition, set its send state variable to the received N(R), stop the P-bit Timer, and resend any unacknowledged PDU's.

If, while in the timer recovery condition, the LLC correctly receives a valid I-format or S-format PDU with the P/F bit set to "O" and with a N(R) within the range from its current send state variable to X included, the LLC shall not clear the timer recovery condition. The LLC shall update the send state variable V(S) for that data link connection to equal the N(R) that was received in the valid I-format or S-format PDU.

If the P-bit Timer runs out in the timer recovery condition, the LLC shall add one to its retransmission count variable. If the retransmission count variable is not equal to N2, the LLC shall resend an S-format PDU with the P bit set to "1" and restart its P-bit Timer.

If the retransmission count variable is equal to N2, the LLC shall initiate a resetting procedure (by sending a SABME command PDU) as described in 7.6 below. N2 is a system parameter (see 7.8.2).

NOTE: Although the LLC may implement the internal variable X, other mechanisms do exist that achieve the identical function.

7.6 Procedures for Resetting. The resetting phase is used to initialize both directions of information transfer according to the procedure described below. The resetting phase shall only apply during the asynchronous balanced mode ABM.

Either LLC shall be able to initiate a resetting of both directions by sending an SABME command PDU and starting its Acknowledgment Timer.

After receiving an SABME command PDU, the LLC shall return, at the earliest opportunity;

(1) a UA response PDU and reset its send and receive state variables V(S) and V(R) to 0 to reset the data link connection, or

(2) a DM response PDU if the data link connection is to terminated.

The return of the UA or DM response PDU shall take precedence over any other

response PDU for the same data link connection which may be pending at the LLC. It shall be possible to follow the UA PDU with additional LLC PDU's, if pending. If the UA PDU is received correctly by the initiating LLC, it shall reset its send and receive state variables V(S) and V(R) to 0 and stop its Acknowledgment Timer. This shall also clear all exception conditions which might be present at either of the LLC's involved in the reset. This exchange shall also indicate clearance of any busy condition that may have been present at either LLC involved in the reset.

If a DM response PDU is received, the LLC shall enter the data link disconnected phase, shall stop its Acknowledgment Timer and shall report to the higher layer for appropriate action. If the Acknowledgment Timer runs out before a UA or DM response PDU is received, the SABME command PDU shall be resent and the Acknowledgment Timer shall be started. After the timer runs out N2 times, the sending LLC shall stop sending the SABME command PDU, shall report to the higher layer for the appropriate error recovery actions to initiate, and shall enter the asynchronous disconnected mode. The value of N2 is defined in 7.8.2.

Other Type 2 PDU's (with the exception of the SABME and DISC command PDU's) which are received by the LLC before completion of the reset procedure shall be discarded.

Under certain FRMR exception conditions listed in 7.7, it shall be possible for the LLC to ask the remote LLC to reset the data link connection by sending a FRMR response PDU.

Upon reception of an FRMR response PDU (even during a FRMR exception condition) the LLC shall initiate a resetting procedure by sending a SABME command PDU, or shall initiate a disconnect procedure by sending a DISC command PDU.

After sending a FRMR response PDU, the LLC shall enter the FRMR exception condition. The FRMR exception condition shall be cleared when the LLC receives or sends an SABME or DISC command PDU or DM response PDU. Any other Type 2 command PDU received while in the FRMR exception condition shall cause the LLC to resend the FRMR response PDU with the same information field as originally sent.

In the FRMR exception condition, additional I PDU's shall not be sent, and received I-format PDU's and S-format PDUs shall be discarded by the LLC.

It shall be possible for the LLC to start its Acknowledgment Timer on the sending of the FRMR response PDU. If the timer runs out before the reception of an SABME or DISC command PDU from the remote LLC, it shall be possible for the LLC to resend the FRMR response PDU and restart its Acknowledgment Timer. After the Acknowledgment Timer has run out N2 times, the LLC shall reset the data link connection by sending a SABME command PDU. The value of N2 is defined in 7.8.2.

When an additional FRMR response PDU is sent while the Acknowledgment Timer is running, the timer shall not be reset or restarted.

7.7 FRMR Exception Conditions. The LLC shall request a resetting proce-

dure (by sending a FRMR response PDU) as described in 7.6, when receiving, during the information transfer phase, a PDU with one of the conditions identified in 5.4.2.3.5. The coding of the information field of the FRMR response PDU which is sent is given in 5.4.2.3.5.

The LLC shall initiate a resetting procedure (by sending a SABME command PDU) as described in 7.6 when receiving a FRMR response PDU during the information transfer phase.

7.8 List of Data Link Connection Parameters. A number of data link connection parameters are defined, the range of values for which are determined on a system-by-system basis by the user at the time that the local area network is established.

The data link connection parameters for Type 2 operation shall be as follows:

7.8.1 Timer Functions. In Type 2 operation it is possible for a number of independent events to be taking place on a data link connection that could each employ a timing function. These timing functions are defined below, as identified in the text that describes Type 2 operation. It is understood that these timing functions can be realized by using a number of individual timers, or by using a single timer. If a single timing function is employed, it will be necessary for the designer to determine on an instance-by-instance basis when to reset and restart the timer and when to let it continue running based on the priority assigned to the individual actions that are in progress.

The periods of the timer functions shall take into account whether the timers are started at the beginning or the end of the event that initiated the timer (eg, sending of a PDU by the LLC), and any delay introduced by the MAC sublayer.

The proper operation of the procedure shall require that the value of the timing functions be greater than the maximum time between the normal network operation of Type 2 PDU's and the reception of the corresponding Type 2 PDU returned as an answer to the initiating Type 2 PDU.

7.8.1.1 Acknowledgment Timer. The Acknowledgment Timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive an acknowledgment to one or more outstanding I PDUs or an expected response PDU to a sent unnumbered command PDU.

7.8.1.2 P-bit Timer. The P-bit Timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a PDU with the F bit set to "1" in response to a sent Type 2 command with the P-bit set to "1."

7.8.1.3 Reject Timer. The Reject (REJ) Timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a reply to a sent REJ PDU.

7.8.1.4 Busy-State Timer. The Busy-State Timer is a data link connection parameter that shall define the time interval during which the LLC shall wait for an indication of the clearance of a busy condition at the other LLC.

7.8.2 Maximum Number of Transmissions N2. N2 is a data link connection parameter that indicates the maximum number of times that a PDU is sent following the running out of the Acknowledgment Timer, the P-bit Timer, or the Reject Timer.

7.8.3 Maximum Number of Octets in an I PDU N1. N1 is a data link connection parameter that denotes the maximum number of octets in an I PDU. Refer to the various MAC descriptions to determine the precise value of N1 for a given medium access method. LLC itself places no restrictions on the value of N1. However, in the interest of having a value for N1 that all users of Type 2 LLC may depend upon, all MAC's must at least be capable of accommodating I PDU's with information fields up to and including 128 octets in length.

7.8.4 Maximum Number of Outstanding I PDU's k. The maximum number (k) of sequentially numbered I PDU's that the LLC may have outstanding (ie, unacknowledged) at any given time shall be a data link connection parameter which can never exceed 127.

7.8.5 Minimum Number of Octets in a PDU. A minimal length valid data link connection PDU shall contain exactly two address fields and one control field in that order. Thus the minimum number of octets in a valid data link connection PDU shall be 3 or 4, depending on whether the PDU is a U-format PDU, or an I-format or S-format, respectively.

7.9 Formal Description of the Type 2 Procedures. If discrepancies appear to exist with the text found in the balance of Section 7, this subsection (7.9) shall be viewed as being the definitive description.

7.9.1 Connection Service Component Overview. The Connection Service Component handles all LLC Type 2 PDU traffic for a specific link connection (designated by a DA, DSAP - SA, SSAP pair). Once activated the Connection Service Component shall process Type 2 LLC PDU's addressed to the local Service Access Point from the remote Service Access Point and shall send Type 2 PDU's to the remote Service Access Point as a result of either a Service Access Point user request or as the result of some data link protocol action. (See Fig 7-1 and Table 7-1.)

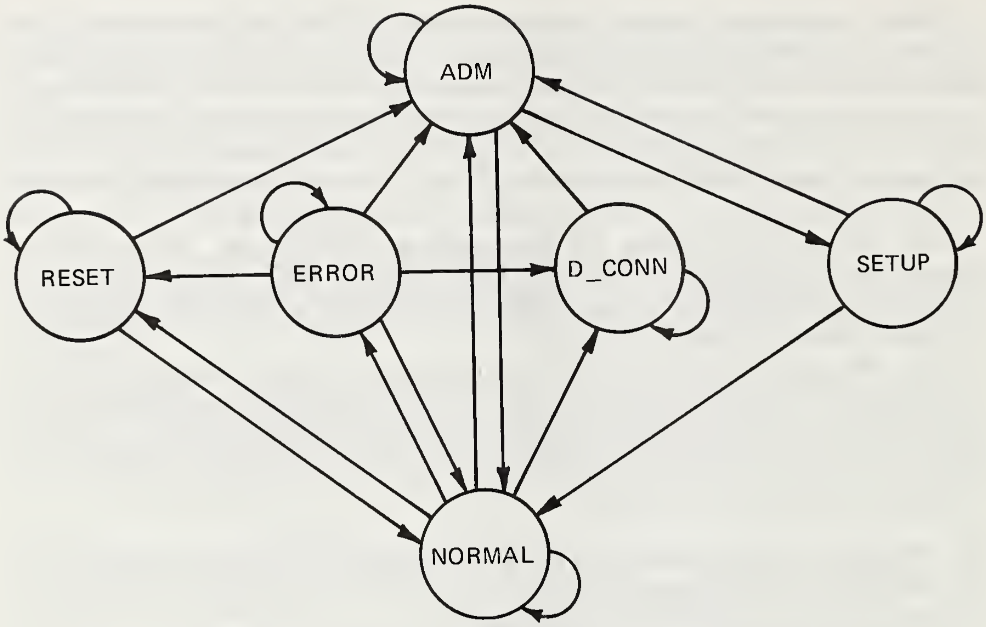
When the Service Access Point Component (as described in 6.9) is placed in the ACTIVE state, all the Connection Service Components associated with the Service Access Point are placed in the ADM (asynchronous disconnected mode) state. When the Service Access Point Component leaves the ACTIVE state, all of the associated Connection Service Components are deactivated, regardless of the current state of the Connection Service Component.

The following points apply to the interpretation of the state tables:

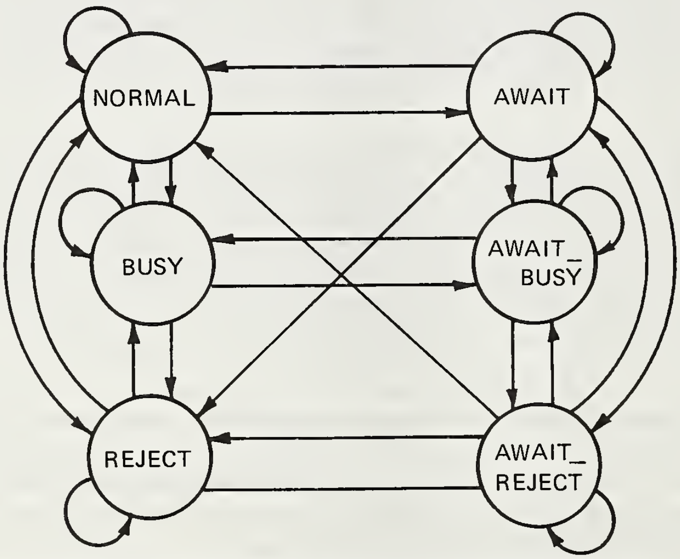
(1) The use of flag variables has been introduced to limit the number of states. The flags maintain the state of particular conditions affecting the connection component.

(2) In the list of events, events of the form RECEIVE_XXX_YYY are listed. The interpretation is that this event is the reception of any command PDU or response PDU not specifically listed for that state.

(3) For some combinations of state and event(s) the tables provide several optional actions. The option selected is done on the basis of (a) local status, (b) the result of layer management action, or (c) implementation decision. There is no relationship between the order of options between events, nor is it implied that the same option must be selected every time the event occurs.



(a) Data Link Establishment, Disconnection and Resetting States



(b) Information Transfer (Connected) States

Fig 7-1
Connection Component State Diagram

Table 7-1
Connection Component State Transitions

Current State	Event	Action(s)	Next State
ADM	CONNECT_REQUEST	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0 S_FLAG:=0 .... CONNECT_CONFIRM(IMPOSSIBLE)	SETUP ...... ADM
	RECEIVE_SABME_CMD(P=X)	CONNECT_INDICATE SEND_UA_RSP(F=P) V(S):=0 V(R):=0 P_FLAG:=0 REMOTE_BUSY:=0 .... SEND_DM_RSP(F=P)	NORMAL ...... ADM
	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P)	ADM
	RECEIVE_XXX_CMD(P=1)	SEND_DM_RSP(F=1)	ADM
	RECEIVE_XXX_CMD(P=0) or RECEIVE_XXX_RSP(F=X)		ADM
SETUP	RECEIVE_SABME_CMD(P=X)	V(S):=0 V(R):=0 SEND_UA_RSP(F=P) S_FLAG:=1	SETUP
	RECEIVE_UA_RSP(F=0) and P_FLAG=0 or RECEIVE_UA_RSP(F=1) and P_FLAG=1	STOP_ACK_TIMER V(S):=0 V(R):=0 UPDATE_P_FLAG CONNECT_CONFIRM(CONNECT) REMOTE_BUSY:=0	NORMAL
	ACK_TIMER_EXPIRED and S_FLAG=1	P_FLAG:=0 CONNECT:=CONFIRM(CONNECT) REMOTE_BUSY:=0	NORMAL
	RECEIVE_DISC_CMD(P=X)	SEND_DM_RSP(F=P) CONNECT_CONFIRM(DISCONNECT) STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X)	CONNECT_CONFIRM(DISCONNECT) STOP_ACK_TIMER	ADM
	RECEIVE_XXX_YYY		SETUP

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
SETUP	ACK_TIMER_EXPIRED and RETRY_COUNT<N2 and S_FLAG=0	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	SETUP
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and S_FLAG=0	CONNECT_CONFIRM(FAILED)	ADM
RESET	RECEIVE_SABME_CMD(P=X)	V(S):=0 V(R):=0 S_FLAG:=1 SEND_UA_RSP(F=P)	RESET
	RECEIVE_UA_RSP(F=0) and P_FLAG=0 and CAUSE_FLAG=1 or RECEIVE_UA_RSP(F=1) and P_FLAG=1 and CAUSE_FLAG=1	STOP_ACK_TIMER V(S):=0 V(R):=0 UPDATE_P_FLAG RESET_CONFIRM(CONNECT) REMOTE_BUSY:=0	NORMAL
	RECEIVE_UA_RSP(F=0) and P_FLAG=0 and CAUSE_FLAG=0 or RECEIVE_UA_RSP(F=1) and P_FLAG=1 and CAUSE_FLAG=0	STOP_ACK_TIMER V(S):=0 V(R):=0 UPDATE_P_FLAG REPORT_STATUS(RESET_DONE) REMOTE_BUSY:=0	NORMAL
	ACK_TIMER_EXPIRED and S_FLAG=1	P_FLAG:=0 REPORT_STATUS(RESET_DONE) REMOTE_BUSY:=0	NORMAL
	RECEIVE_DISC_CMD(P=X) and CAUSE_FLAG=1	SEND_DM_RSP(F=P) RESET_CONFIRM(DISCONNECT) STOP_ACK_TIMER	ADM
	RECEIVE_DISC_CMD(P=X) and CAUSE_FLAG=0	SEND_DM_RSP(F=P) REPORT_STATUS(RESET_REFUSED) STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X) and CAUSE_FLAG=1	RESET_CONFIRM(DISCONNECT) STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X) and CAUSE_FLAG=0	REPORT_STATUS(RESET_REFUSED) STOP_ACK_TIMER	ADM
	RECEIVE_XXX_YYY		RESET

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
RESET	DATA_CONN_REQUEST	DATA_CONN_CONFIRM(REFUSE)	RESET
	ACK_TIMER_EXPIRED and RETRY_COUNT<N2 and S_FLAG=0	SEND_SABME_CMD(P_X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	RESET
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and CAUSE_FLAG=1 and S_FLAG=0	RESET_CONFIRM(FAILED)	ADM
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and CAUSE_FLAG=0 and S_FLAG=0	REPORT_STATUS(RESET_FAILED)	ADM
D_CONN	RECEIVE_SABME_CMD(P=X) and CAUSE_FLAG=1	SEND_DM_RSP(F=P) DISCONNECT_CONFIRM(CONFLICT) STOP_ACK_TIMER	ADM
	RECEIVE_SABME_CMD(P=X) and CAUSE_FLAG=0	SEND_DM_RSP(F=P) REPORT_STATUS(CONFLICT) STOP_ACK_TIMER	ADM
	RECEIVE_UA_RSP(F=0) and P_FLAG=0 and CAUSE_FLAG=1 or RECEIVE_UA_RSP(F=1) and P_FLAG=1 and CAUSE_FLAG=1	STOP_ACK_TIMER DISCONNECT_CONFIRM(DISCONNECT)	ADM
	RECEIVE_UA_RSP(F=0) and P_FLAG=0 and CAUSE_FLAG=0 or RECEIVE_UA_RSP(F=1) and P_FLAG=1 and CAUSE_FLAG=0	STOP_ACK_TIMER REPORT_STATUS(DISCONNECT)	ADM
	RECEIVE_DISC_CMD(P=X)	SEND_UA_RSP(F=P)	D_CONN
	RECEIVE_DM_RSP(F=X) and CAUSE_FLAG=1	STOP_ACK_TIMER DISCONNECT_CONFIRM(DISCONNECT)	ADM
	RECEIVE_DM_RSP(F=X) and CAUSE_FLAG=0	STOP_ACK_TIMER REPORT_STATUS(DISCONNECT)	ADM
	RECEIVE_XXX_YYY		D_CONN

**Table 7-1 (cont'd)
Connection Component State Transitions**

Current State	Event	Action(s)	Next State
D_CONN	DATA_CONN_REQUEST	DATA_CONN_CONFIRM(REFUSE)	D_CONN
	ACK_TIMER_EXPIRED and RETRY_COUNT<N2	SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	D_CONN
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and CAUSE_FLAG=1	DISCONNECT_CONFIRM(FAILED)	ADM
	ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 and CAUSE_FLAG=0	REPORT_STATUS(FAILED)	ADM
ERROR	RECEIVE_SABME_CMD(P=X)	V(S):=0 V(R):=0 SEND_UA_RSP(F=P) RESET_INDICATE P_FLAG:=0 REMOTE_BUSY:=0 STOP_ACK_TIMER	NORMAL
	RECEIVE_DISC_CMD(P=X)	SEND_UA_RSP(F=P) DISCONNECT_INDICATE STOP_ACK_TIMER	ADM
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATE STOP_ACK_TIMER	ADM
	RECEIVE_FRMR_RSP(F=X)	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0 REPORT_STATUS(FRMR_RECEIVED) CAUSE_FLAG:=0 .... SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0 REPORT_STATUS(FRMR_RECEIVED) CAUSE_FLAG:=0	RESET ...... D_CONN
	RECEIVE_XXX_CMD(P=X)	RE-SEND_FRMR_RSP(F=P)	ERROR
	RECEIVE_XXX_RSP(F=X)		ERROR
	DATA_CONN_REQUEST	DATA_CONN_CONFIRM(REFUSE)	ERROR

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
ERROR	ACK_TIMER_EXPIRED and RETRY_COUNT<N2	RE-SEND_FRMR_RSP(F=0) START_ACK_TIMER RETRY_COUNT:=RETRY_COUNT+1	ERROR
	ACK_TIMER_EXPIRED and RETRY_COUNT>N2	SEND_SABME_CMD(P=X) S_FLAG:=0 P_FLAG:=P START_ACK_TIMER RETRY_COUNT:=0 REPORT_STATUS(RESETTING) CAUSE_FLAG:=0	RESET
NORMAL or BUSY or REJECT or AWAIT or AWAIT_BUSY or AWAIT_REJECT	DISCONNECT_REQUEST	SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0 CAUSE_FLAG:=1	D_CONN
	RESET_REQUEST	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0 CAUSE_FLAG:=1	RESET
	RECEIVE_SABME_CMD(P=X)	V(S):=0 V(R):=0 SEND_UA_RSP(F=P) RESET_INDICATE P_FLAG:=0 REMOTE_BUSY:=0 STOP_ALL_TIMERS .... SEND_DM_RSP(F=P) REPORT_STATUS(RESET_DECLINED) STOP_ALL_TIMERS	NORMAL ...... ADM

**Table 7-1 (cont'd)
Connection Component State Transitions**

Current State	Event	Action(s)	Next State
NORMAL or BUSY or REJECT or AWAIT or AWAIT_ BUSY or AWAIT_ REJECT	RECEIVE_DISC_CMD(P=X)	SEND_UA_RSP(F=P) DISCONNECT_INDICATE STOP_ALL_TIMERS	ADM
	RECEIVE_FRMR_RSP(F=X)	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0 REPORT_STATUS(FRMR_RECEIVED) CAUSE_FLAG:=0 .... SEND_DISC_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0 REPORT_STATUS(FRMR_RECEIVED) CAUSE_FLAG:=0	RESET ...... D_CONN
	RECEIVE_DM_RSP(F=X)	DISCONNECT_INDICATE STOP_ALL_TIMERS	ADM
	RECEIVE_ZZZ_CMD(P=X)_ WITH_INVALID_N(R) or RECEIVE_I_CMD(P=X)_ WITH_INVALID_N(S)	SEND_FRMR_RSP(F=P) REPORT_STATUS(FRMR_SENT) START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0	ERROR
	RECEIVE_ZZZ_RSP(F=X)_ WITH_INVALID_N(R) or RECEIVE_I_RSP(F=X)_ WITH_INVALID_N(S) or RECEIVE_UA_RSP(F=X) or RECEIVE_XXX_RSP(F=1) and P_FLAG=0 or RECEIVE_BAD_PDU	SEND_FRMR_RSP(F=0) REPORT_STATUS(FRMR_SENT) START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0	ERROR
	P_TIMER_EXPIRED and RETRY_COUNT>=N2 or ACK_TIMER_EXPIRED and RETRY_COUNT>=N2 or REJ_TIMER_EXPIRED and RETRY_COUNT>=N2 or BUSY_TIMER_EXPIRED and RETRY_COUNT>=N2	SEND_SABME_CMD(P=X) P_FLAG:=P START_ACK_TIMER STOP_OTHER_TIMERS RETRY_COUNT:=0 REPORT_STATUS(RESETTING) CAUSE_FLAG:=0	RESET

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
NORMAL	DATA_CONN_REQUEST and REMOTE_BUSY=0 and P_FLAG=0	SEND_I_CMD(P=1) START_P_TIMER START_ACK_TIMER_ IF_NOT_RUNNING	NORMAL ......
		SEND_I_XXX(X=0) START_ACK_TIMER_ IF_NOT_RUNNING .... DATA_CONN_CONFIRM(REFUSE)	NORMAL ...... NORMAL
	DATA_CONN_REQUEST and REMOTE_BUSY=0 and P_FLAG=1	SEND_I_XXX(X=0) START_ACK_TIMER_ IF_NOT_RUNNING .... DATA_CONN_CONFIRM(REFUSE)	NORMAL ...... NORMAL
	DATA_CONN_REQUEST and REMOTE_BUSY=1	DATA_CONN_CONFIRM(REMOTE_BUSY)	NORMAL
	LOCAL_BUSY_DETECTED and P_FLAG=0	SEND_RNR_CMD(P=1) START_P_TIMER DATA_FLAG:=0 .... SEND_RNR_XXX(X=0) DATA_FLAG:=0	BUSY ...... BUSY
	LOCAL_BUSY_DETECTED and P_FLAG=1	SEND_RNR_XXX(X=0) DATA_FLAG:=0	BUSY
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=0 or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=0 or RECEIVE_I_RSP(F=1)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	SEND_REJ_XXX(X=0) UPDATE_N(R) UPDATE_P_FLAG START_REJ_TIMER IF_F=1_CLEAR_REMOTE_BUSY .... SEND_REJ_CMD(P=1) UPDATE_N(R) START_P_TIMER IF_F=1_CLEAR_REMOTE_BUSY	REJECT ...... REJECT
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=1 or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	SEND_REJ_XXX(X=0) UPDATE_N(R) START_REJ_TIMER	REJECT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
NORMAL	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_REJ_RSP(F=1) UPDATE_N(R) START_REJ_TIMER	REJECT
	RECEIVE_I_RSP(F=0) and P_FLAG=0 or RECEIVE_I_CMD(P=0) and P_FLAG=0 or RECEIVE_I_RSP(F=1) and P_FLAG=1	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_ACKNOWLEDGE_CMD(P=1) START_P_TIMER UPDATE_N(R) IF_F=1_CLEAR_REMOTE_BUSY	NORMAL ......
		V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_P_FLAG SEND_ACKNOWLEDGE_XXX(X=0) UPDATE_N(R) IF_F=1_CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_I_RSP(F=0) and P_FLAG=1 or RECEIVE_I_CMD(P=0) and P_FLAG=1	V(R):=V(R)=+1 DATA_CONN_INDICATE(INFORMATION) SEND_ACKNOWLEDGE_XXX(X=0) UPDATE_N(R)	NORMAL
	RECEIVE_I_CMD(P=1)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_ACKNOWLEDGE_RSP(F=1) UPDATE_N(R)	NORMAL
	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_RR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R) CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_RR_CMD(P=1)	SEND_ACKNOWLEDGE_RSP(F=1) UPDATE_N(R) CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0) or RECEIVE_RNR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R) SET_REMOTE_BUSY	NORMAL

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
NORMAL	RECEIVE_RNR_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R) SET_REMOTE_BUSY	NORMAL
	RECEIVE_REJ_CMD(P=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=1) and P_FLAG=1	V(S):=N(R) UPDATE_N(R) UPDATE_P_FLAG RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY .... V(S):=N(R) UPDATE_N(R) START_P_TIMER RE-SEND_I_CMD(P=1) CLEAR_REMOTE_BUSY	NORMAL ...... NORMAL
	RECEIVE_REJ_CMD(P=0) and P_FLAG=1 or RECEIVE_REJ_RSP(F=0) and P_FLAG=1	V(S):=N(R) UPDATE_N(R) RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_REJ_CMD(P=1)	V(S):=N(R) UPDATE_N(R) RE-SEND_I_RSP(F=1) CLEAR_REMOTE_BUSY	NORMAL
	INITIATE_P/F_CYCLE and P_FLAG=0	SEND_RR_CMD(P=1) START_P_TIMER	NORMAL
	P_TIMER_EXPIRED and_RETRY_COUNT<N2	P_FLAG:=0 .... SEND_RR_CMD(P=1) RESET_V(S) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1	NORMAL ...... AWAIT
	ACK_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2 or BUSY_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2	SEND_RR_CMD(P=1) RESET_V(S) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1	AWAIT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
BUSY	DATA_CONN_REQUEST and REMOTE_BUSY=0 and P_FLAG=0	SEND_I_CMD(P=1) START_P_TIMER START_ACK_TIMER_IF_NOT_RUNNING .... SEND_I_XXX(X=0) START_ACK_TIMER_IF_NOT_RUNNING .... DATA_CONN_CONFIRM(REFUSE)	BUSY ...... BUSY ...... BUSY
	DATA_CONN_REQUEST and REMOTE_BUSY=0 and P_FLAG=1	SEND_I_XXX(X=0) START_ACK_TIMER_IF_NOT_RUNNING .... DATA_CONN_CONFIRM(REFUSE)	BUSY ...... BUSY
	DATA_CONN_REQUEST and REMOTE_BUSY=1	DATA_CONN_CONFIRM(REMOTE_BUSY)	BUSY
	LOCAL_BUSY_CLEARED and DATA_FLAG=1 and P_FLAG=0	SEND_REJ_CMD(P=1) START_REJ_TIMER START_P_TIMER .... SEND_REJ_XXX(X=0) START_REJ_TIMER	REJECT ...... REJECT
	LOCAL_BUSY_CLEARED and DATA_FLAG=1 and P_FLAG=1	SEND_REJ_XXX(X=0) START_REJ_TIMER	REJECT
	LOCAL_BUSY_CLEARED and DATA_FLAG=0 and P_FLAG=0	SEND_RR_CMD(P=1) START_P_TIMER .... SEND_RR_XXX(X=0)	NORMAL ...... NORMAL
	LOCAL_BUSY_CLEARED and DATA_FLAG=0 and P_FLAG=1	SEND_RR_XXX(X=0)	NORMAL
	LOCAL_BUSY_CLEARED and DATA_FLAG=2 and P_FLAG=0	SEND_RR_CMD(P=1) START_P_TIMER .... SEND_RR_XXX(X=0)	REJECT ...... REJECT
	LOCAL_BUSY_CLEARED and DATA_FLAG=2 and P_FLAG=1	SEND_RR_XXX(X=0)	REJECT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
BUSY	RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=0 or RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=0 or RECEIVE_I_RSP(F=1)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_P_FLAG UPDATE_N(R) IF_DATA_FLAG=0_THEN_ DATA_FLAG:=1 IF_F=1_CLEAR_REMOTE_BUSY .... SEND_RNR_CMD(P=1) START_P_TIMER UPDATE_N(R) IF_DATA_FLAG=0_THEN_ DATA_FLAG:=1 IF F=1 CLEAR REMOTE BUSY	BUSY ...... BUSY
	RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=1 or RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) IF_DATA_FLAG=0_THEN_ DATA_FLAG:=1	BUSY
	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_RNR_RSP(F=1) UPDATE_N(R) IF_DATA_FLAG=0_THEN_ DATA FLAG:=1	BUSY
	RECEIVE_I_CMD(P=1)	SEND_RNR_RSP(F=1) UPDATE_N(R) IF_DATA_FLAG=2_ STOP_REJ_TIMER DATA_FLAG:=1 .... V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_RNR_RSP(F=1) UPDATE_N(R) IF_DATA_FLAG=2_ STOP_REJ_TIMER DATA_FLAG:=0	BUSY ...... BUSY

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
BUSY	RECEIVE_I_RSP(F=0) and P_FLAG=0 or RECEIVE_I_CMD(P=0) and P_FLAG=0 or RECEIVE_I_RSP(F=1) and P_FLAG=1	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_P_FLAG UPDATE_N(R) IF_DATA_FLAG=2_ STOP_REJ_TIMER DATA_FLAG:=1 IF_F=1_CLEAR_REMOTE_BUSY	BUSY
		SEND_RNR_CMD(P=1) START_P_TIMER UPDATE_N(R) IF_DATA_FLAG=2_ STOP_REJ_TIMER DATA_FLAG:=1 IF_F=1_CLEAR_REMOTE_BUSY	BUSY
		V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_RNR_CMD(P=1) START_P_TIMER UPDATE_N(R) IF_DATA_FLAG=2 STOP_REJ_TIMER DATA_FLAG:=0 IF_F=1_CLEAR_REMOTE_BUSY	BUSY
		V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_P_FLAG OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) IF_DATA_FLAG=2 STOP_REJ_TIMER DATA_FLAG:=0 IF_F=1_CLEAR REMOTE BUSY	BUSY
	RECEIVE_I_RSP(F=0) and P_FLAG=1 or RECEIVE_I_CMD(P=0) and P_FLAG=1	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) IF_DATA_FLAG=2_ STOP_REJ_TIMER DATA_FLAG:=1	BUSY
		V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) IF_DATA_FLAG=2_ STOP_REJ_TIMER DATA_FLAG:=0	BUSY

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
BUSY	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_RR_RSP(F=1) and P FLAG=1	UPDATE_P_FLAG UPDATE_N(R) CLEAR_REMOTE_BUSY	BUSY
	RECEIVE_RR_CMD(P=1)	SEND_RNR_RSP(F=1) UPDATE_N(R) CLEAR_REMOTE_BUSY	BUSY
	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0) or RECEIVE_RNR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R) SET_REMOTE_BUSY	BUSY
	RECEIVE_RNR_CMD(P=1)	SEND_RNR_RSP(F=1) UPDATE_N(R) SET_REMOTE_BUSY	BUSY
	RECEIVE_REJ_CMD(P=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=1) and P_FLAG=1	V(S):=N(R) UPDATE_N(R) UPDATE_P_FLAG RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY .... V(S):=N(R) UPDATE_N(R) RE-SEND_I_CMD(P=1) START_P_TIMER CLEAR_REMOTE_BUSY	BUSY ...... BUSY
	RECEIVE_REJ_CMD(P=0) and P_FLAG=1 or RECEIVE_REJ_RSP(F=0) and P_FLAG=1	V(S):=N(R) UPDATE_N(R) RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	BUSY
	RECEIVE_REJ_CMD(P=1)	V(S):=N(R) UPDATE_N(R) SEND_RNR_RSP(F=1) RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	BUSY
	INITIATE_P/F_CYCLE and P_FLAG=0	SEND_RNR_CMD(P=1) START_P_TIMER	BUSY

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
BUSY	P_TIMER_EXPIRES and RETRY_COUNT<N2	P_FLAG:=0 .... SEND_RNR_CMD(P=1) RESET_V(S) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1	BUSY ...... AWAIT_ BUSY
	ACK_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2 or BUSY_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2	SEND_RNR_CMD(P=1) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1 RESET_V(S)	AWAIT_ BUSY
	REJ_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2	DATA_FLAG:=1 .... SEND_RNR_CMD(P=1) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1 RESET_V(S) DATA_FLAG:=1	BUSY ...... AWAIT_ BUSY
	REJ_TIMER_EXPIRES and P_FLAG-1 and RETRY_COUNT<N2	DATA_FLAG:=1	BUSY
REJECT	DATA_CONN_REQUEST and REMOTE_BUSY=0 and P_FLAG=0	SEND_I_CMD(P=1) START_P_TIMER .... SEND_I_XXX(X=0) .... DATA_CONN_CONFIRM(REFUSE)	REJECT ...... REJECT ...... REJECT
	DATA_CONN_REQUEST and REMOTE_BUSY=0 and P_FLAG=1	SEND_I_XXX(X=0) .... DATA_CONN_CONFIRM(REFUSE)	REJECT ...... REJECT
	DATA_CONN_REQUEST and REMOTE_BUSY=1	DATA_CONN_CONFIRM(REMOTE_BUSY)	REJECT
	LOCAL_BUSY_DETECTED and P_FLAG=0	SEND_RNR_CMD(P=1) START_P_TIMER DATA_FLAG:=2 .... SEND_RNR_XXX(X=0) DATA_FLAG:=2	BUSY ...... BUSY
	LOCAL_BUSY_DETECTED and P_FLAG=1	SEND_RNR_XXX(X=0) DATA_FLAG:=2	BUSY

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
REJECT	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S) or RECEIVE_I_RSP(F=1)_ WITH_UNEXPECTED_N(S) and P_FLAG=1	UPDATE_N(R) UPDATE_P_FLAG IF_F=1_CLEAR_REMOTE_BUSY	REJECT
	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_RR_RSP(F=1) UPDATE_N(R)	REJECT
	RECEIVE_I_RSP(F=0) and P_FLAG=0 or RECEIVE_I_CMD(P=0) and P_FLAG=0 or RECEIVE_I_RSP(F=1) and P_FLAG=1	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_ACKNOWLEDGE_CMD(P=1) START_P_TIMER UPDATE_N(R) IF_F=1_CLEAR_REMOTE_BUSY STOP_REJ_TIMER	NORMAL ......
		V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_P_FLAG SEND_ACKNOWLEDGE_XXX(X=0) UPDATE_N(R) IF_F=1_CLEAR_REMOTE_BUSY STOP_REJ_TIMER	NORMAL
	RECEIVE_I_RSP(F=0) and P_FLAG=1 or RECEIVE_I_CMD(P=0) and P_FLAG=1	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_ACKNOWLEDGE_XXX(X=0) UPDATE_N(R) STOP_REJ_TIMER	NORMAL
	RECEIVE_I_CMD(P=1)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_ACKNOWLEDGE_RSP(F=1) UPDATE_N(R) STOP_REJ_TIMER	NORMAL
	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_RR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R) CLEAR_REMOTE_BUSY	REJECT
	RECEIVE_RR_CMD(P=1)	SEND_ACKNOWLEDGE_RSP(F=1) UPDATE_N(R) CLEAR_REMOTE_BUSY	REJECT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
REJECT	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0) or RECEIVE_RNR_RSP(F=1) and P_FLAG=1	UPDATE_P_FLAG UPDATE_N(R) SET_REMOTE_BUSY	REJECT
	RECEIVE_RNR_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R) SET_REMOTE_BUSY	REJECT
	RECEIVE_REJ_CMD(P=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=0) and P_FLAG=0 or RECEIVE_REJ_RSP(F=1) and P_FLAG=1	V(S):=N(R) UPDATE_N(R) UPDATE_P_FLAG RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY .... V(S):=N(R) UPDATE_N(R) RE-SEND_I_CMD(P=1) START_P_TIMER CLEAR_REMOTE_BUSY	REJECT ...... REJECT
	RECEIVE_REJ_CMD(P=0) and P_FLAG=1 or RECEIVE_REJ_RSP(F=0) and P_FLAG=1	V(S):=N(R) UPDATE_N(R) RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	REJECT
	RECEIVE_REJ_CMD(P=1)	V(S):=N(R) UPDATE_N(R) RE-SEND_I_RSP(F=1) CLEAR_REMOTE_BUSY	REJECT
	INITIATE_P/F_CYCLE and P_FLAG=0	SEND_RR_CMD(P=1) START_P_TIMER	REJECT
	REJ_TIMER_EXPIRED and P-FLAG=0 and RETRY_COUNT<N2	SEND_REJ_CMD(P=1) START_P_TIMER START_REJ_TIMER RETRY_COUNT:=RETRY_COUNT+1 .... EMPTY	REJECT ...... NORMAL
	P_TIMER_EXPIRED and RETRY_COUNT<2	P_FLAG:=0 .... SEND_RR_CMD(P=1) START_P_TIMER START_REJ_TIMER RETRY_COUNT:=RETRY_COUNT+1 RESET_V(S)	REJECT ...... AWAIT_ REJECT

**Table 7-1 (cont'd)
Connection Component State Transitions**

Current State	Event	Action(s)	Next State
REJECT	ACK_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2 or BUSY_TIMER_EXPIRED and P_FLAG=0 and RETRY_COUNT<N2	SEND_RR_CMD(P=1) START_P_TIMER START_REJ_TIMER RETRY_COUNT:=RETRY_COUNT+1 RESET_V(S)	AWAIT_ REJECT
AWAIT	DATA_CONN_REQUEST	DATA_CONN_CONFIRM(REFUSE)	AWAIT
	LOCAL_BUSY_DETECTED	SEND_RNR_XXX(X=0) DATA_FLAG:=0	AWAIT_ BUSY
	RECEIVE_I_RSP(F=1) WITH_UNEXPECTED_N(S)	SEND_REJ_XXX_(X=0) UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER RE-SEND_I_XXX(X=0) START_REJ_TIMER CLEAR_REMOTE_BUSY	REJECT
		....	......
		SEND_REJ_CMD(P=1) UPDATE_N(R) UPDATE_V(S) RE-SEND_I_XXX(X=0) START_P_TIMER START_REJ_TIMER CLEAR_REMOTE_BUSY	REJECT
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S)	SEND_REJ_XXX(X=0) UPDATE_N(R) UPDATE_V(S) START_REJ_TIMER	AWAIT_ REJECT
	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_REJ_RSP(F=1) UPDATE_N(R) UPDATE_V(S) START_REJ_TIMER START_P_TIMER	AWAIT_ REJECT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
AWAIT	RECEIVE_I_RSP(F=1)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_N(R) UPDATE_V(S) RE-SEND_I_CMD(P=1)_OR_SEND_RR START_P_TIMER CLEAR_REMOTE_BUSY .... V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) STOP_P_TIMER UPDATE_N(R) UPDATE_V(S) RE-SEND_I_XXX(X=0)_OR_SEND_RR CLEAR_REMOTE_BUSY	NORMAL ...... NORMAL
	RECEIVE_I_RSP(P=0) or RECEIVE_I_CMD(P=0)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_RR_XXX(X=0) UPDATE_N(R) UPDATE_V(S)	AWAIT
	RECEIVE_I_CMD(P=1)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_RR_RSP(F=1) UPDATE_N(R) UPDATE_V(S)	AWAIT
	RECEIVE_RR_RSP(F=1) or RECEIVE_REJ_RSP(F=1)	UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	NORMAL
	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_REJ_CMD(P=0) or RECEIVE_REJ_RSP(F=0)	UPDATE_N(R) UPDATE_V(S) CLEAR_REMOTE_BUSY	AWAIT
	RECEIVE_RR_CMD(P=1) or RECEIVE_REJ_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) CLEAR_REMOTE_BUSY	AWAIT
	RECEIVE_RNR_RSP(F=1)	UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER SET_REMOTE_BUSY	NORMAL

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
AWAIT	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0)	UPDATE_N(R) UPDATE_V(S) SET_REMOTE_BUSY	AWAIT
	RECEIVE_RNR_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) SET_REMOTE_BUSY	AWAIT
	P_TIMER_EXPIRED and RETRY_COUNT<N2	SEND_RR_CMD(P=1) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1	AWAIT
AWAIT_ BUSY	DATA_CONN_REQUEST	DATA_CONN_CONFIRM(REFUSE)	AWAIT_ BUSY
	LOCAL_BUSY_CLEARED and DATA_FLAG=1	SEND_REJ_XXX(X=0) START_REJ_TIMER	AWAIT_ REJECT
	LOCAL_BUSY_CLEARED and DATA_FLAG=0	SEND_RR_XXX(X=0)	AWAIT
	LOCAL_BUSY_CLEARED and DATA_FLAG=2	SEND_RR_XXX(X=0)	AWAIT_ REJECT
	RECEIVE_I_RSP(F=1)_ WITH_UNEXPECTED_N(S)	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER DATA_FLAG:=1 CLEAR_REMOTE_BUSY RE-SEND_I_XXX(X=0) .... SEND_RNR_CMD(P=1) UPDATE_N(R) UPDATE_V(S) START_P_TIMER DATA_FLAG:=1 CLEAR_REMOTE_BUSY RE-SEND_I_XXX(X=0)	BUSY ...... BUSY
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S)	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=1	AWAIT_ BUSY
	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_RNR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=1	AWAIT_ BUSY

**Table 7-1 (cont'd)
Connection Component State Transitions**

Current State	Event	Action(s)	Next State
AWAIT_ BUSY	RECEIVE_I_RSP(F=1)	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=1 STOP_P_TIMER CLEAR_REMOTE_BUSY RE-SEND_I_XXX(X=0)	BUSY ......
		SEND_RNR_CMD(P=1) V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) START_P_TIMER UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=0 CLEAR_REMOTE_BUSY RE-SEND_I_XXX(X=0)	BUSY ......
		OPTIONAL_SEND_RNR_XXX(X=0) V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) STOP_P_TIMER UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=0 CLEAR_REMOTE_BUSY RE-SEND_I_XXX(X=0)	BUSY
	RECEIVE_I_RSP(F=0) or RECEIVE_I_CMD(P=0)	OPTIONAL_SEND_RNR_XXX(X=0) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=1	AWAIT_ BUSY ......
		OPTIONAL_SEND_RNR_XXX(X=0) V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=0	AWAIT_ BUSY
	RECEIVE_I_CMD(P=1)	SEND_RNR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=1	AWAIT_ BUSY ......
		SEND_RNR_RSP(F=1) V(R)=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_N(R) UPDATE_V(S) DATA_FLAG:=0	AWAIT_ BUSY

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
AWAIT_ BUSY	RECEIVE_RR_RSP(F=1) or RECEIVE_REJ_RSP(F=1)	UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY	BUSY
	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_REJ_CMD(P=0) or RECEIVE_REJ_RSP(F=0)	UPDATE_N(R) UPDATE_V(S) CLEAR_REMOTE_BUSY	AWAIT_ BUSY
	RECEIVE_RR_CMD(P=1) or RECEIVE_REJ_CMD(P=1)	SEND_RNR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) CLEAR_REMOTE_BUSY	AWAIT_ BUSY
	RECEIVE_RNR_RSP(F=1)	UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER SET_REMOTE_BUSY	BUSY
	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0)	UPDATE_N(R) UPDATE_V(S) SET_REMOTE_BUSY	AWAIT_ BUSY
	RECEIVE_RNR_CMD(P=1)	SEND_RNR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) SET_REMOTE_BUSY	AWAIT_ BUSY
	P_TIMER_EXPIRED and RETRY_COUNT<N2	SEND_RNR_CMD(P=1) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1	AWAIT_ BUSY
AWAIT_ REJECT	DATA_CONN_REQUEST	DATA_CONN_CONFIRM(REFUSE)	AWAIT_ REJECT
	LOCAL_BUSY_DETECTED	SEND_RNR_XXX(X=0) DATA_FLAG:=2	AWAIT_ BUSY
	RECEIVE_I_CMD(P=0)_ WITH_UNEXPECTED_N(S) or RECEIVE_I_RSP(F=0)_ WITH_UNEXPECTED_N(S)	UPDATE_N(R) UPDATE_V(S)	AWAIT_ REJECT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
AWAIT_ REJECT	RECEIVE_I_CMD(P=1)_ WITH_UNEXPECTED_N(S)	SEND_RR_RSP(F=1) UPDATE_V(R) UPDATE_V(S)	AWAIT_ REJECT
	RECEIVE_I_RSP(F=1)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) UPDATE_N(R) UPDATE_V(S) RE-SEND_I_CMD(P=1)_OR_SEND_RR START_P_TIMER STOP_REJ_TIMER CLEAR_REMOTE_BUSY .... V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) STOP_P_TIMER STOP_REJ_TIMER UPDATE_N(R) UPDATE_V(S) RE-SEND_I_XXX(X=0)_OR_SEND_RR CLEAR_REMOTE_BUSY	NORMAL ...... NORMAL
	RECEIVE_I_RSP(F=0) or RECEIVE_I_CMD(P=0)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_RR_XXX(X=0) STOP_REJ_TIMER UPDATE_N(R) UPDATE_V(S)	AWAIT
	RECEIVE_I_CMD(P=1)	V(R):=V(R)+1 DATA_CONN_INDICATE(INFORMATION) SEND_RR_RSP(F=1) STOP_REJ_TIMER UPDATE_N(R) UPDATE_V(S)	AWAIT
	RECEIVE_RR_RSP(F=1) or RECEIVE_REJ_RSP(F=1) or RECEIVE_I_RSP(F=1) WITH_UNEXPECTED_N(S)	UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER RE-SEND_I_XXX(X=0) CLEAR_REMOTE_BUSY .... UPDATE_N(R) UPDATE_V(S) RE-SEND_I_CMD(P=1) START_P_TIMER CLEAR_REMOTE_BUSY	REJECT ...... REJECT

Table 7-1 (cont'd)
Connection Component State Transitions

Current State	Event	Action(s)	Next State
AWAIT_REJECT	RECEIVE_RR_CMD(P=0) or RECEIVE_RR_RSP(F=0) or RECEIVE_REJ_CMD(P=0) or RECEIVE_REJ_RSP(F=0)	UPDATE_N(R) UPDATE_V(S) CLEAR_REMOTE_BUSY	AWAIT_REJECT
	RECEIVE_RR_CMD(P=1) or RECEIVE_REJ_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) CLEAR_REMOTE_BUSY	AWAIT_REJECT
	RECEIVE_RNR_RSP(F=1)	UPDATE_N(R) UPDATE_V(S) STOP_P_TIMER SET_REMOTE_BUSY	REJECT
	RECEIVE_RNR_CMD(P=0) or RECEIVE_RNR_RSP(F=0)	UPDATE_N(R) UPDATE_V(S) SET_REMOTE_BUSY	AWAIT_REJECT
	RECEIVE_RNR_CMD(P=1)	SEND_RR_RSP(F=1) UPDATE_N(R) UPDATE_V(S) SET_REMOTE_BUSY	AWAIT_REJECT
	P_TIMER_EXPIRED and RETRY_COUNT<N2	SEND_REJ_CMD(P=1) START_P_TIMER RETRY_COUNT:=RETRY_COUNT+1	AWAIT_REJECT

(4) In the list of actions there is no implied ordering, unless one or more of the actions is conditional upon the value(s) of flag(s) which are modified by other actions. In this case the test(s) must be completed before the flag(s) are modified.

(5) In the list of actions, actions of the form SEND_xxx_RSP (F=1) are indicated. It should be noted that if some other response PDU (with the F bit set to "0") will be sent earlier, it is permissible to modify that PDU from E bit set to "0" to F bit set to "1," and to send the new PDU with the F bit set to "0." This could occur, for example, if an LLC implementation managed the queue of PDU's awaiting transmission.

(6) For simplicity the state table has four timers: the ACK_TIMER for timing acknowledgments, the P_TIMER for timing the P/F cycle, the REJ_TIMER for timing the "sent REJ" condition, and the BUSY_TIMER for timing the "remote busy" condition. It should be noted that by the addition of appropriate flags a functionally equivalent state table can be developed that requires only one timer.

(7) Any START_TIMER action (re)starts the specified timer from zero, even if the timer is already running. When the time reaches its limit the appropriate TIMER_EXPIRED condition is set and the timer stopped. The TIMER _EXPIRED condition is cleared when it is recognized by the Connection Component state machine. The STOP_TIMER action stops the timer if it is running or clears the TIMER_EXPIRED condition if the timer has already reached its limit.

NOTE: To insure proper interpretation of the state table entries, the descriptions of the entries (see 7.9.1.1—7.9.1.3) should be read in concert with the state tables.

7.9.1.1 Connection Component State Descriptions.

(1) **ADM.** The connection component is in the asynchronous disconnected mode. It can accept a SABME PDU from a remote LLC SSAP or, at the request of the Service Access Point user, can initiate an SABME PDU transmission to a remote LLC DSAP, to establish a data link connection. It also responds to a DISC command PDU and a command PDU with the P bit set to "1."

(2) **SETUP.** The connection component has transmitted a SABME command PDU to a remote LLC DSAP and is waiting for a reply.

(3) **NORMAL.** A data link connection exists between the local LLC Service Access Point and the remote LLC Service Access Point. Sending and reception of information and supervisory PDU's can be performed.

(4) **BUSY.** A data link connection exists between the local LLC Service Access Point and the remote LLC Service Access Point. I PDU's may be sent. Local conditions make it likely that the information field of received I PDU's will be ignored. Supervisory PDU's may be both sent and received.

(5) **REJECT.** A data link connection exists between the local LLC Service Access Point and the remote LLC Service Access Point. The local connection component has requested that the remote connection component resend a specific I PDU that the local connection component has detected as being out of sequence. Both I PDU's and supervisory PDU's may be sent and received.

(6) **AWAIT.** A data link connection exists between the local LLC Service Access Point and the remote LLC Service Access Point. The local LLC is performing a timer recovery operation and has sent a command PDU with the P bit set to "1," and is awaiting an acknowledgment from the remote LLC. I PDU's may be received but not sent. Supervisory PDU's may be both sent and received.

(7) **AWAIT_BUSY.** A data link connection exists between the local LLC Service Access Point and the remote LLC Service Access Point. The local LLC is performing a timer recovery operation and has sent a command PDU with the P bit set to "1," and is awaiting an acknowledgment from the remote LLC. I PDU's may not be sent. Local conditions make it likely that the information field of received I PDU's will be ignored. Supervisory PDU's may be both sent and received.

(8) **AWAIT_REJECT.** A data link connection exists between the local LLC Service Access Point and the remote LLC Service Access Point. The local connection component has requested that the remote connection component re-transmit a specific I PDU that the local connection component has detected as being out of sequence. Before the local LLC entered this state it was performing a

timer recovery operation and had sent a command PDU with the P bit set to "1," and is still awaiting an acknowledgment from the remote LLC. I PDU's may be received but not transmitted. Supervisory PDU's may be both transmitted and received.

(9) **D_CONN.** At the request of the Service Access Point user the local LLC has sent a DISC command PDU to the remote LLC DSAP and is waiting for a reply.

(10)**RESET.** As a result of a Service Access Point user request or the receipt of a FRMR response PDU the local connection component has sent a SABME command PDU to the remote LLC DSAP to reset the data link connection and is waiting for a reply.

(11) **ERROR.** The local connection component has detected an error in a received PDU and has sent a FRMR response PDU. It is waiting for a reply from the remote connection component.

7.9.1.2 Connection Service Component Event Description. In the list of events below the value of the P or F bits in received commands and responses is listed as X. In the state transition tables values of 0, 1, or X or are used. The latter indicates that either 0 or 1 may occur in the event.

(1) **CONNECT_REQUEST.** The user has requested that a data link connection be established with a remote LLC DSAP.

(2) **DATA_CONN_REQUEST.** The user has requested that a data unit be sent to the remote LLC DSAP.

(3) **DISCONNECT_REQUEST.** The user has requested that the data link connection with the remote LLC DSAP be terminated.

(4) **RESET_REQUEST.** The user has requested that the data link connection with the remote LLC DSAP be reset.

(5) **LOCAL_BUSY_DETECTED.** The local station has entered a busy condition and may not be able to accept I PDU's from the remote LLC SSAP.

(6) **LOCAL_BUSY_CLEARED.** The local station busy condition has ended and the station can accept I PDU's from the remote LLC SSAP.

(7) **RECEIVE_BAD_PDU.** The remote SSAP has sent a command or response PDU addressed to the local DSAP which is not implemented, or has an information field when not permitted or is an I PDU with an information field length greater than can be accommodated by the local LLC.

(8) **RECEIVE_DISC_CMD(P=X).** The remote SSAP has sent a DISC command PDU with the F bit set to "X" addressed to the local DSAP.

(9) **RECEIVE_DM_RSP(F=X).** The remote SSAP has sent a DM response PDU with the F bit set to "X" addressed to the local DSAP.

(10) **RECEIVE_FRMR_RSP(F=X).** The remote SSAP has sent a FRMR response PDU with The F bit set to "X" addressed to the local DSAP.

(11) **RECEIVE_I_CMD(P=X).** The remote SSAP has sent a I command PDU with the P bit set to "X" addressed to the local DSAP. Both the N(R) and N(S) field are valid and the N(S) value is the expected sequence number.

(12) **RECEIVE_I_CMD(–=X)_WITH_UNEXPECTED_N(S).** The remote SSAP has sent an I command PDU with the P bit set to "X" addressed to the

local DSAP. The N(S) field of the command does not contain the expected sequence number but is within the window size. The N(R) field is valid.

(13) **RECEIVE_I_CMD(P=X)_WITH_INVALID_N(S).** The remote SSAP has sent an I command PDU with the P bit set to "X" addressed to the local SSAP. The N(S) field of the command is invalid. The N(R) field is valid.

(14) **RECEIVE_I_RSP(F=X).** The remote SSAP has sent a I response PDU with the F bit set to "X" addressed to the local DSAP. Both the N(R) and N(S) field are valid and the N(S) value is the expected sequence number.

(15) **RECEIVE_I_RSP(F=X) _WITH_UNEXPECTED_N(S).** The remote SSAP has sent an I response PDU with the F bit set to "X" addressed to the local DSAP. The N(S) field of the command does not contain the expected sequence number but is within the window size.

(16) **RECEIVE_I_RSP(F=X)_WITH_INVALID_N(S).** The remote SSAP has sent an I response PDU with the F bit set to "X" addressed to the local DSAP. The N(S) field of the response is invalid. The N(R) field is valid.

(17) **RECEIVE_REJ_CMD(P=X).** The remote SSAP has sent a REJ command PDU with the P bit set to "X" addressed to the local DSAP.

(18) **RECEIVE_REJ_RSP(F=X).** The remote SSAP has sent a REJ response PDU with the F bit set to "X" addressed to the local DSAP.

(19) **RECEIVE_RNR_CMD(P=X).** The remote SSAP has sent a RNR command PDU with the P bit set to "X" addressed to the local DSAP.

(20) **RECEIVE_RNR_RSP(F=X).** The remote SSAP has sent a RNR response PDU with the F bit set to "X" addressed to the local DSAP.

(21) **RECEIVE_RR_CMD(P=X).** The remote SSAP has sent a RR command PDU with the P bit set to "X" addressed to the local DSAP.

(22) **RECEIVE_RR_RSP(F=X).** The remote SSAP has sent a RR response PDU with the F bit set to "X" addressed to the local DSAP.

(23) **RECEIVE_SABME_CMD(P=X).** The remote SSAP has sent a SABME command PDU with the P bit set to "X" addressed to the local DSAP.

(24) **RECEIVE_UA_RSP(F=X).** The remote SSAP has sent a UA response PDU with the F bit set to "X" addressed to the local DSAP.

(25) **RECEIVE_XXX_CMD(P=X).** The remote SSAP has sent a type 2 command PDU with the P bit set to "X" addressed to the local DSAP. The command is any command not specifically listed for that state.

(26) **RECEIVE_XXX_RSP(F=X).** The remote SSAP has sent a type 2 response PDU with the F bit set to "X" addressed to the local DSAP. The response is any response not specifically listed for that state.

(27) **RECEIVE_XXX_YYY.** The remote SSAP has sent a type 2 PDU addressed to the local DSAP. The PDU is any command or response not specifically listed for that state.

(28) **RECEIVE_ZZZ_CMD(P=X)_WITH_INVALID_N(R).** The remote SSAP has sent an I, RR, RNR or REJ command PDU with the P bit set to "X" addressed to the local DSAP. The N(R) field of the command is invalid.

(29) **RECEIVE_ZZZ_RSP(F=X)_WITH_INVALID_N(R).** The remote SSAP has sent an I, RR, RNR or REJ command PDU with the P bit set to "X" addressed to the local DSAP. The N(R) field of the command is invalid.

(30) **P_TIMER_EXPIRED.** The P/F cycle timer has expired.

(31) **ACK_TIMER_EXPIRED.** The Acknowledgment timer has expired.
(32) **REJ_TIMER_EXPIRED.** The "sent REJ" timer has expired.
(33) **BUSY_TIMER_EXPIRED.** The remote-busy timer has expired.

In the state transition table some of the above events are qualified by the following conditions. The event is recognized only when the condition is true.

(34) **CAUSE_FLAG=1.** When CAUSE_FLAG has a value of one the RESET or D_CONN state was entered as a result of higher layer action.
(35) **CAUSE_FLAG=0.** When CAUSE_FLAG has a value of zero the RESET or D_CONN state was entered as a result of LLC action.
(36) **DATA_FLAG=1.** When DATA_FLAG has a value of one, data unit(s) from I PDU's were discarded during a local busy period.
(37) **DATA_FLAG=0.** When DATA_FLAG has a value of zero, data unit(s) from I PDU's were not discarded during a local busy period.
(38) **DATA_FLAG=2.** When DATA_FLAG has a value of two, the BUSY state was entered from the REJECT state, and the requested I PDU has not yet been received.
(39) **P_FLAG=1.** P_FLAG has a value of one when a command with the P bit set to "1" has been sent and a response with the F bit set to "1" is expected.
(40) **P_FLAG=0.** P_FLAG has a value of zero when a response PDU with the F bit set to "1" is not expected.
(41) **REMOTE_BUSY=1.** When REMOTE_BUSY has a value of one, a RNR PDU has been received from the remote connection component to indicate that I PDU's should not be sent.
(42) **REMOTE_BUSY=0.** When REMOTE_BUSY has a value of zero sending of I PDU is possible.
(43) **RETRY_COUNT<N2.** The number of retries is less than the maximum number of retries.
(44) **RETRY_COUNT>=N2.** The number of retries has reached the maximum number permissible.
(45) **S_FLAG=1.** In the SETUP and RESET states an S_FLAG value of one indicates that a SABME PDU has been received.
(46) **S_FLAG=0.** In the SETUP and RESET states an S_FLAG value of zero indicates an SABME PDU has not been received.

7.9.1.3 Connection Component Action Description. In the list of actions described below the value of the P or F bits in the transmitted commands and responses is listed as X. In the list of state transition table actions given below values of 0, 1 or x are used. The latter indicates that either 0 or 1 may be used.

(1) **CLEAR_REMOTE_BUSY.** If REMOTE_BUSY is one then set REMOTE_BUSY to zero to indicate the remote LLC is now able to accept I PDU's, stop the BUSY_TIMER, inform the user by issuing REPORT_STATUS (REMOTE_NOT_BUSY), and start the (re)sending of any I PDU's that were waiting for the remote busy to be cleared.
(2) **CONNECT_INDICATE.** Inform the user that a connection has been established by a remote LLC SSAP.

(3) **CONNECT_CONFIRM.** The connection component indicates the result of the CONNECT_REQUEST issued earlier by the user. The valid results are:

CONNECT. The remote LLC DSAP accepted the connection.
DISCONNECT. The remote DSAP refused the connection request.
FAILED. The remote DSAP never replied to the connection request.
IMPOSSIBLE. The local station is unable to initiate the connection request, eg, shortage of buffers, etc.

(4) **DATA_CONN_INDICATE(INFORMATION).** The connection service component passes the data unit from the received I PDU to the user.

(5) **DATA_CONN_CONFIRM.** The connection component informs the user of the result of a DATA_CONN_REQUEST issued earlier by the user. The valid responses are:

REFUSE. The connection component did not accept the data unit for transmission. Possible reasons would be shortage of buffers, or the transmission window being full, or timer recovery action due to not receiving acknowledgments for I PDU's sent previously or the connection component is in a state where it can not send I PDU's.
REMOTE_BUSY. The connection component did not accept the data unit for transmission because the remote LLC DSAP is busy and not accepting I PDU's.
RECEIVED. The remote LLC DSAP has acknowledged receipt of one or more data units submitted earlier by the user with a DATA_CONN_REQUEST.

(6) **DISCONNECT_INDICATE.** Inform the user that the remote LLC SSAP has initiated disconnection of the data link connection.

(7) **DISCONNECT_CONFIRM.** The connection component indicates the result of the DISCONNECT_REQUEST issued earlier by the user. The valid results are:

CONFLICT. The remote DSAP was attempting to reset the connection. The connection was disconnected.
DISCONNECT. The remote LLC DSAP accepted the disconnection.
FAILED. The remote DSAP never replied to the disconnection request. The connection was disconnected.

(8) **RESET_INDICATE.** Inform the user that the remote LLC SSAP has initiated a reset of the data link connection.

(9) **RESET_CONFIRM.** The connection component indicates the result of the RESET_REQUEST issued earlier by the user. The valid results are:

CONNECT. The remote LLC DSAP accepted the resetting command.
DISCONNECT. The remote DSAP refused the reset request. The connection was disconnected.

FAILED. The remote DSAP never replied to the reset request. The connection was disconnected.

(10) **REPORT_STATUS.** Report the status of the data link connection to the user. Permissible status values are:

CONFLICT. The local connection component was attempting to disconnect while the remote LLC was attempting a reset. The connection was disconnected.
DISCONNECT. The remote LLC DSAP accepted the disconnection.
FAILED. The remote DSAP never replied to the disconnect request the connection was disconnected.
FRMR_RECEIVED. The local connection component has received a FRMR response PDU.
FRMR_SENT. The local connection component has received an invalid PDU, and has sent a FRMR response PDU.
REMOTE_BUSY. The remote LLC DSAP is busy. The local connection component will not accept a DATA_CONN_REQUEST.
REMOTE_NOT_BUSY. The remote LLC DSAP is no longer busy. The local connection component will now accept a DATA_CONN_REQUEST.
RESETTING. The local connection component has initiated a reset of the data link connection due to the timer retry count reaching its limit.
RESET_DECLINED. The local connection component has refused a reset request from the remote LLC. The connection was disconnected.
RESET_DONE. The remote LLC DSAP accepted the reset request.
RESET_FAILED. The remote DSAP never replied to the reset request. The connection was disconnected.
RESET_REFUSED. The remote LLC refused the reset request. The connection was disconnected.

(11) **IF_F=1_CLEAR_REMOTE_BUSY.** If the I PDU is a response with the F bit set to "1" in response to a command PDU with the P bit set to "1," then perform the CLEAR_REMOTE_BUSY_ action.

(12) **IF_DATA_FLAG=2_STOP_REJ_TIMER.** If DATA_FLAG has a value of two, indicating that a REJ PDU has been sent, stop the "sent REJ" timer.

(13) **INITIATE_PF_CYCLE.** The local LLC wants to initiate a P/F cycle. (This is only required if the local LLC is not generating other command PDU's for some reason.)

(14) **SEND_DISC_CMD(P=X).** Transmit a DISC command PDU with the P bit set to "X" to the remote LLC DSAP.

(15) **SEND_DM_RSP(F=X).** Send a DM response PDU with the F bit set to "X" to the remote LLC DSAP.

(16) **SEND_FRMR_RSP(F=X).** Send a FRMR response PDU with the F bit set to "X" to the remote LLC DSAP.

(17) **RE-SEND_FRMR_RSP(F=0).** Send the same FRMR response PDU as sent earlier to the remote LLC DSAP. Set the F bit to "0."

(18) **RE-SEND_FRMR_RSP(F=P).** Send the same FRMR response PDU

as sent earlier to the remote LLC DSAP. Set the F bit equal to the P bit of the received command PDU.

(19) **SEND_I_CMD(P=1).** Send an I command PDU with the P bit set to "1" to the remote LLC DSAP with the data unit supplied by the user with the DATA_CONN_REQUEST. Before transmission copy the current values of the send state variable V(S) and the receive state variable V(R) into the N(S) and N(R) fields, respectively, of the I PDU and increment (modulo 128) the send state variable V(S).

(20) **RE_SEND_I_CMD(P=1).** Start resending all the unacknowledged I PDU's for this data link connection beginning with the N(R) given in the received PDU. Send the first as a command with the P bit set to "1." If the queue contains more than one I PDU, the balance may be sent as commands with the P bit set to "0", or as responses with the F bit set to "0."

(21) **RE_SEND_I_CMD(P=1)_OR_SEND_RR.** Start resending all the unacknowledged I PDU's for this data link connection beginning with the N(R) given in the received PDU. Send the first as a command with the P bit set to "1." If the queue contains more than one I PDU the balance must be sent as commands with the P bit set to "0" or as responses with the F bit set to "0." It is permissible to send a RR command PDU with the P bit set to "1" to the remote LLC DSAP before starting the resending of the I PDU's. In this case the first I PDU is sent as a command with the P bit set to "0" or as a response with the F bit set to "0." If no I PDU is ready to send a RR command PDU with the P bit set to "1" must be sent to the remote LLC LSAP.

(22) **SEND_I_XXX(X=0).** Send either an I response PDU with the F bit set to "0" or an I command PDU with the P bit set to 0 to the remote LLC DSAP with the data unit supplied by the user with the DATA_CONN_REQUEST. Before transmission copy the current values of the send state variable V(S) and the receive state variable V(R) into the N(S) and N(R) fields, respectively, of the I PDU and increment (modulo 128) the send state variable V(S).

(23) **RE_SEND_I_XXX(X=0).** Start resending all the unacknowledged I PDUs for this data link connection beginning with the N(R) given in the received PDU. They must be sent as either commands with the P bit set to "0" or as responses with the F bit set to "0."

(24) **RE_SEND_I_XXX(X=0)_OR_SEND_RR.** Start resending all the unacknowledged I PDU's for this data link connection beginning with the N(R) given in the received PDU. They must be sent as either commands with the P bit set to "0" or as responses with the F bit set to "0." It is permissible to send either a RR response PDU with the F bit set to "0" or a RR command PDU with the P bit set to "0" to the remote LLC DSAP before starting the resending of the I PDU's. If no I PDU is ready to send either an RR response PDU with the F bit set to "0" or an RR command PDU with the P bit set to "0" must be sent to the remote LLC DSAP.

(25) **RE-SEND_I_RSP(F=1).** Start resending all the unacknowledged I PDU's for this data link connection beginning with the N(R) given in the received PDU. Send the first as a response with the F bit set to "1." If the queue contains more than one I PDU the balance must be transmitted as commands with the P bit set to "0" or as responses with the F bit set to "0."

(26) **SEND_REJ_CMD(P=1).** Send a REJ command PDU with the P bit set to "1" to the remote LLC DSAP.

(27) **SEND_REJ_RSP(F=1).** Send a REJ response PDU with the F bit set to "1" to the remote LLC DSAP.

(28) **SEND_REJ_XXX(X=0).** Send either a REJ response PDU with the F bit set to "0" or a REJ command PDU with the P bit set to "0" to the remote LLC DSAP.

(29) **SEND_RNR_CMD(P=1).** Send a RNR command PDU with the P bit set to "1" to the remote LLC DSAP.

(30) **SEND_RNR_RSP(F=1).** Send a RNR response PDU with the F bit set to "1" to the remote LLC DSAP.

(31) **SEND_RNR_XXX(X=0).** Send either a RNR response PDU with the F bit set to "0" or a RNR command PDU with the P bit set to "0" to the remote LLC DSAP.

(32) **SET_REMOTE_BUSY.** If REMOTE_BUSY is zero, then set REMOTE_BUSY to one to indicate the remote LLC is in the busy state and is not able to accept I PDU's, start the BUSY_TIMER, inform the user by issuing REPORT_STATUS (REMOTE_BUSY) and stop any (re)sending of I PDU's that is in progress.

(33) **OPTIONAL_SEND_RNR_XXX_(X=0).** It is permissible to send a RNR command PDU with the P bit set to "0" or a RNR response PDU with the F bit set to "0" to the remote LLC DSAP in case the remote LLC did not receive the first RNR sent when the busy state was entered.

(34) **SEND_RR_CMD(P=1).** Send a RR command PDU with the P bit set to "1" to the remote LLC DSAP.

(35) **SEND_ACKNOWLEDGE_CMD(P=1).** Under all conditions it is permissible to send a RR command PDU with the P bit set to "1" to the remote LLC DSAP. If no I PDU is ready to send the RR command PDU with the P bit set to "1" must be sent to the remote LLC DSAP. (This RR PDU may be delayed, by a time bounded by the timer T1 value, to wait for the generation of an I PDU.) However, if an I PDU is ready to send, and can be modified to a command with the P bit set to "1," then the RR command PDU does not need to be sent.

(36) **SEND_RR_RSP(F=1).** Send a RR response PDU with the F bit set to "1" to the remote LLC DSAP.

(37) **SEND_ACKNOWLEDGE_RSP(F=1).** Under all conditions it is permissible to send a RR response PDU with the F bit set to "1" to the remote LLC DSAP. If no I PDU is ready to send the RR response PDU with the F bit set to "1" must be sent to the remote LLC DSAP. However, if an I PDU is ready to send, and can be modified to a response with the F bit set to "1," then the RR response PDU does not need to be sent.

(38) **SEND_RR_XXX(X=0).** Send either a RR response PDU with the F bit set to "0" or a RR command PDU with the P bit set to "0" to the remote LLC DSAP.

(39) **SEND_ACKNOWLEDGE_XXX(X=0).** Under all conditions it is permissible to send either a RR response PDU with the F bit set to "0" or a RR command PDU with the P bit set to "0" to the remote LLC DSAP.

If no I PDU is ready to send either an RR response with the F bit set to "0" or an RR command PDU with the P bit set to "0" must be sent to the remote LLC DSAP. (This RR PDU may be delayed, by a time bounded by the timer T1 value, to wait for the generation of an I PDU.) However, if an I PDU is ready to send then the RR PDU does not need to be sent.

(40) **SEND_SABME_CMD(P=X).** Send a SABME command PDU with the P bit set to "X" to the remote LLC DSAP.

(41) **SEND_UA_RSP(F=X).** Send a UA response PDU with the F bit set to "X" to the remote LLC DSAP.

(42) **S_FLAG:=0.** Set S_FLAG to zero to indicate that a SABME PDU has not been received from the remote LLC while the local connection component is in the RESET or SETUP state.

(43) **S_FLAG:=1.** Set S_FLAG to one to indicate that a SABME PDU has been received from the remote LLC while the local connection component is in the RESET or SETUP state.

(44) **START_P_TIMER.** Start the P/F cycle timer from zero and set P_FLAG to one.

(45) **START_ACK_TIMER.** Start the acknowledgment timer from zero.

(46) **START_REJ_TIMER.** Start the "sent REJ" timer from zero.

(47) **START_ACK_TIMER_IF_NOT_RUNNING.** If the acknowledgment timer is not currently running then start the acknowledgment timer from zero.

(48) **STOP_ACK_TIMER.** Stop the acknowledgment timer.

(49) **STOP_P_TIMER.** Stop the P/F cycle timer and set P flag to 0.

(50) **STOP_REJ_TIMER.** Stop the "sent REJ" timer.

(51) **STOP_ALL_TIMERS.** Stop the P/F cycle timer, the "sent REJ" timer, the remote busy timer, and the acknowledgment timer.

(52) **STOP_OTHER_TIMERS.** Stop the P/F cycle timer, the "sent REJ" timer and the remote busy timer.

(53) **UPDATE_N(R).** If the N(R) of the received PDU acknowledges the receipt of one or more previously unacknowledged I PDU's update the local record of N(R), set RETRY_COUNT to zero, stop the acknowledgment timer and inform the user by issuing DATA_CONN_CONFIRM(RECEIVED). If unacknowledged I PDU's still exist, start the acknowledgment timer if it was stopped. (Note: If some form of SEND_I PDU is initiated at the same time as UPDATE_N(R), then the acknowledgment timer is always started if it was stopped.)

(54) **UPDATE_P_FLAG.** If the received PDU was a response with the F bit set to "1" set the P_FLAG to zero and stop the P/F cycle timer.

(55) **CAUSE_FLAG:=0.** Set the CAUSE_FLAG to zero to record that the RESET or D_CONN state was entered due to LLC action.

(56) **CAUSE_FLAG:=1.** Set the CAUSE_FLAG to one to record that the RESET or D_CONN state was entered due to higher layer action.

(57) **DATA_FLAG:=2.** Set the DATA_FLAG to two to record that the BUSY state was entered with a REJ PDU outstanding.

(58) **DATA_FLAG:=0.** Set the DATA_FLAG to zero to indicate that the data units from received I PDU's were not discarded during a local busy period.

(59) **DATA_FLAG:=1.** Set the DATA_FLAG to one to indicate that the data units from received I PDU's were discarded during a local busy period.

(60) **IF_DATA_FLAG=0_THEN_DATA_FLAG:=1.** If the DATA_FLAG had been zero indicating that no data units had been ignored set it to one to indicate that data units have now been ignored.

(61) **P_FLAG:=0.** Initialize the P_FLAG to zero. This indicates that the reception of a response PDU with the F bit set to "1" is not expected.

(62) **P_FLAG:=P.** Set the P_FLAG to the value of the P bit in the command PDU being sent.

(63) **REMOTE_BUSY:=0.** Set REMOTE_BUSY to zero to indicate that the remote LLC is able to accept I PDU's.

(64) **RETRY_COUNT:=0.** Initialize RETRY_COUNT to zero.

(65) **RETRY_COUNT:=RETRY_COUNT+1.** Increment RETRY_COUNT by one.

(66) **V(R):=0.** Initialize the receive state variable. This is the expected sequence number of the next I PDU received.

(67) **V(R):=V(R)+1.** Increment (modulo 128) the receive state variable. This is the expected sequence number of the next I PDU received.

(68) **V(S):=0.** Initialize the send state variable. This is the sequence number of the next I PDU to be sent.

(69) **V(S):=N(R).** Reset the send state variable to the value specified by the N(R) field of the REJ PDU just received.

(70) **RESET_V(S).** Set the variable X to the current value of V(S), and set V(S) to the last N(R) received from the remote LLC.

(71) **UPDATE_V(S).** If the N(R) of the received PDU lies in the range from the current V(S) to X, set V(S) to the value of N(R).